

進化への自発的な発見を促すリアルタイム骨格比較ツールの開発

著者	岑村 春香
内容記述	筑波大学修士(情報学)学位論文 ・ 平成29年3月24日授与(37770号)
発行年	2017
URL	http://hdl.handle.net/2241/00150803

進化への自発的な発見を促す
リアルタイム骨格比較ツールの開発

筑波大学
図書館情報メディア研究科
2017 年 3 月
岑村 春香

目次

第1章 はじめに

1.1	研究背景	1
1.1.1	リアルタイム CG の普及	1
1.1.2	ICT 教育の普及とデジタル教材	4
1.1.3	進化教材の不足と必要性	6
1.2	本研究のねらい	8

第2章 先行研究・先行事例

2.1	進化教材の開発へ向けた取り組み	9
2.2	形態比較を目的とした 3DCG の活用	11
2.3	WebGL を利用したデジタル教材	14
2.4	タッチデバイスの教育への活用	16

第3章 提案手法

3.1	提案手法の概要	18
3.2	提案の目的	18
3.3	進化学習を目的とした設計	19
3.3.1	従来型の進化教材：前肢の骨格比較	19
3.3.2	3DCG の変形表現	20
3.3.3	系統樹を使用した比較対象の選択	21
3.4	想定する使用環境	21

第4章 リアルタイム骨格比較ツールの開発

4.1	開発環境	23
4.2	題材の選定	24
4.3	3DCG モデル変形表現の実現	24
4.4	レンダリング順の制御	28
4.5	画面構成と機能	30
4.5.1	画面構成	30
4.5.2	3D ビューの操作方法	30
4.5.3	系統樹型 UI による比較対象の選択	32
4.5.4	変形アニメーションの再生	33
4.5.5	閲覧支援機能	33
4.5.5.1	カメラトラッキング	33
4.5.5.2	テクスチャによる配色方法の切り替え	35
4.6	操作性に関する予備評価実験	36
4.7	変更点および追加機能	36

第5章 評価実験	
5.1 実験の目的	37
5.2 実験方法	37
5.2.1 被験者	37
5.2.2 実験手法	37
5.3 実験結果	39
5.3.1 教材内容に関するテスト	39
5.3.2 発見的学習に関する自由記述アンケート	40
5.3.3 印象・操作性に関するアンケート・感想	43
第6章 考察	
6.1 発見的学習の支援	45
6.2 閲覧による知識の定着	46
6.3 学習者の感情	47
6.4 操作性	48
6.5 機能	49
第7章 まとめ	
7.1 結論	50
7.2 展望	51
謝辞	52
参考文献	53
付録	55

図目次

図 1-1	WebGL による高精細 3DCG モデルの描画[16]	3
図 1-2	WebGL によるリアルタイム CG アニメーション[10]	3
図 1-3	公立学校のタブレット端末配備台数	4
図 1-4	系統樹の例 ⁴	7
図 2-1	TimeTree: The Timescale of Life の検索結果例	10
図 2-2	3DCG モデルのモーフィング処理の概略図	12
図 2-3	ウマとヒトの前肢における骨格の対応関係	13
図 2-4	変形アニメーションの様子	13
図 2-5	天体シミュレーション教材の実行画面	15
図 2-6	中井らによるタッチ操作	17
図 3-1	前肢の骨格比較[18]	20
図 3-2	系統樹型 UI	22
図 4-1	モデルファイルの制作	25
図 4-2	制作した CG モデル	26
図 4-3	変形アニメーションの様子(ヒトとコウモリ)	27
図 4-4	骨格構造に差がある場合の変形表現	28
図 4-5	レンダリング順の制御による重畳表示	29
図 4-6	画面構成	31
図 4-7	系統樹型 UI の選択に応じた動作	32
図 4-8	カメラモード	34
図 4-9	2 種類の配色方法	35

表目次

表 4.1	開発環境	23
表 5.1	テストの結果(*:p<0.1).....	39
表 5.2	5段階評価によるアンケート結果	43

第1章 はじめに

1.1 研究背景

1.1.1 リアルタイム CG の普及

リアルタイム CG とは

昨今、3次元コンピュータグラフィックス(3DCG)は、ハードウェア・ソフトウェアの両面で目覚ましい発展を遂げている。3DCG は映画からテレビ、ゲーム、CM など様々なメディアで使用されており、すでに視覚文化に深く根付いている。

2次元のデータしか持たない写真や絵と違い、3DCG はシーンと呼ばれる3次元上の仮想空間にオブジェクトを構築していくことで表現される。シーン内に立体的な構造を持つ3DCG モデルを設置し、モデルの質感情報や空間内の光源を設定後、仮想的なカメラによってそれらを撮影することで画像を生成している。このように、質感情報や光源を元に画像を生成し、2次元情報として出力する工程をレンダリングと呼ぶ。空間上に配置された3DCG モデルは、3次元情報を持つ頂点と面によって構成されているため、カメラに対するモデルの位置を変更することで、現実存在している物と同様に、あらゆる方向から見る事が可能になる。

また、シーン内に配置されたオブジェクトやカメラ、モデルの頂点位置を制御し、少しずつ異なった画像を連続してレンダリングすることで、様々なアニメーションを作成できる。このような3DCG の性質を生かし、ユーザーの操作などの情報入力に対してシーン内の状態を変更し、即座に映像を生成したものがリアルタイム CG である。高速にレンダリングを繰り返すリアルタイム CG は、3DCG のインタラクティブ性を飛躍的に高め、情報表現の幅を大きく広げている。

リアルタイム CG は、CG 計算などの画面描画処理を高速に行う GPU(Graphics Processing Unit)によって実現される。ハードウェアの進歩により、従来は大型コンピュータでなければ不可能だったフォトリアルな CG 表現が、GPU を搭載した一般のコンピュータでリアルタイムに動作するようになった。

しかしハードウェアの技術が進歩する一方で、ソフト面において3DCG のリアルタイム表示を実現するためには、OpenGL や Direct3D などの専用ライブラリと、C や C++などのプログラミング言語を用いた単独のアプリケーションを作成する必要がある。これらの3DCG アプリケーションは、コンパイルされた実行形式のバイナリコードであるため、開発環境と異なる環境では動かない場合もある。また、閲覧環境を整えるためには、ほとんどの場合アプリケーションやゲームエンジンのインストールや、ライブラリの用意が必要となる。

また、リアルタイム CG はゲームを始めとして広く利用されているが、近年まで Web アプリケーションへの進出が遅れていた。Web ブラウザは、ますます高速になり、安定し、セキュリティが強化されていったが、リアルタイム CG を作成できないという短所を有していた。このことから、長年の間、リアルタイム CG の描画にはネイティブアプリケーションが使用されていた。

Web アプリケーションへの進出

近年まで Web アプリケーションへの進出が遅れていたリアルタイム CG であるが、その状況を一変したのが、2011 年 3 月に公開された WebGL である。WebGL は、Web ブラウザ上で高度な 3DCG を描画するための規格である。

WebGL は、JavaScript から OpenGL 3.2 もしくは OpenGL ES 2.0 の関数を呼び出す API(Application Programming Interface)として設計されている。WebGL は、HTML(Hyper Text Markup Language)の最新版¹である HTML5 の canvas 要素を用いることで、Web ブラウザ上にピクセル単位で描画される。この技術を使用することで、Web アプリケーションにおいてもリアルタイム CG の実現が可能となった。

Anyuru[1]や松田[25]は WebGL の長所として、次のような特徴を挙げている。

- ・ オープンスタンダードであるため、誰でも無料で利用できる
- ・ デバイスに内蔵された GPU を活用できるため、描画処理が高速である
- ・ 実行に必要なものは Web ブラウザのみであり、プラグインを必要としない
- ・ Web ページとして作成されるため、Web ブラウザの機能を使用できる
- ・ Web ブラウザ上で動作するため、OS などの実行環境の違いに合わせてプログラムを変える必要がない

上記のような長所は、リアルタイム CG アプリケーションの開発をより効率化するとともに、リアルタイム CG を描画するまでのプロセスをより容易なものとしている。

WebGL によって、精細な 3DCG モデル (図 1-1)や CG アニメーションのリアルタイム表示(図 1-2)が実現されたことで、Web アプリケーションは新たな表現手段を獲得したと言える。

WebGL をサポートする Web ブラウザは、発表以降徐々に数を増していき、2017 年 1 月現在では、Internet Explorer, Google Chrome, Safari といった主要な Web ブラウザの最新版全てが、PC・モバイル用共に WebGL に対応している。

今後、リアルタイム CG を利用した Web アプリケーションは増加していく可能性が高く、Web ブラウザの表現の幅が 3 次元まで広がっていくことが予想される。

¹ 2017 年 1 月現在



図 1-1 WebGL による高精細 3DCG モデルの描画[16]

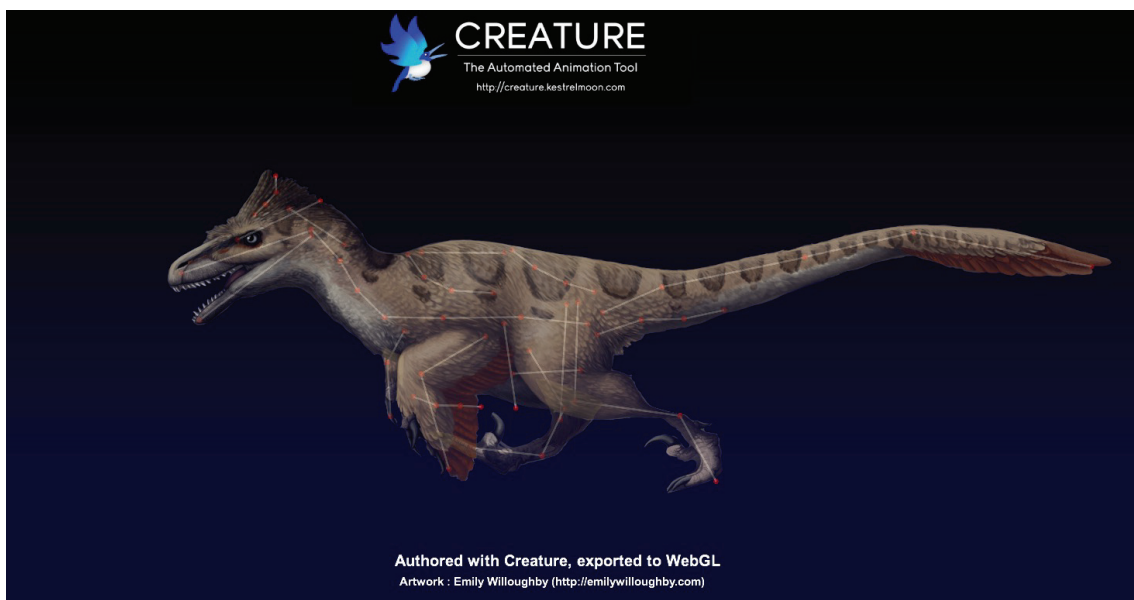


図 1-2 WebGL によるリアルタイム CG アニメーション[10]

1.1.2 ICT 教育の普及とデジタル教材

ICT 教育の普及

情報化の進展は CG や Web 技術のみならず、教育の現場へも影響を与えている。

2010 年からタブレット端末市場の急成長に伴い世界各国の教育現場では、紙媒体の教科書などをデジタル端末に置き変える動きが活発になっている。

日本では、平成 23 年 4 月に文部科学省が、2020 年度に向けた教育の情報化に関する総合的な制作方針である「教育の情報化ビジョン[8]」を策定し、小・中・高等学校及び特別支援学校における教育の情報化事業を推進してきた。その中で政府は、児童・生徒に 1 人 1 台タブレット端末を支給し、日々の学習や授業に活用する計画を打ち出している。教育の現場における情報化は全国で進んでおり、平成 28 年 10 月に文部科学省によって公表された「平成 27 年度 学校における教育の情報化の実態等に関する調査結果[24]」では、全国の公立学校におけるタブレット端末の台数が急激に増加していることが示された(図 1-3)。調査が開始された平成 24 年 3 月以降、タブレット端末の台数は毎年前年度を上回る速度で上昇しており、平成 26 年 3 月からの 2 年間で、その配備数が 72,678 台から 253,755 台へと、実に約 3.5 倍に増加している。

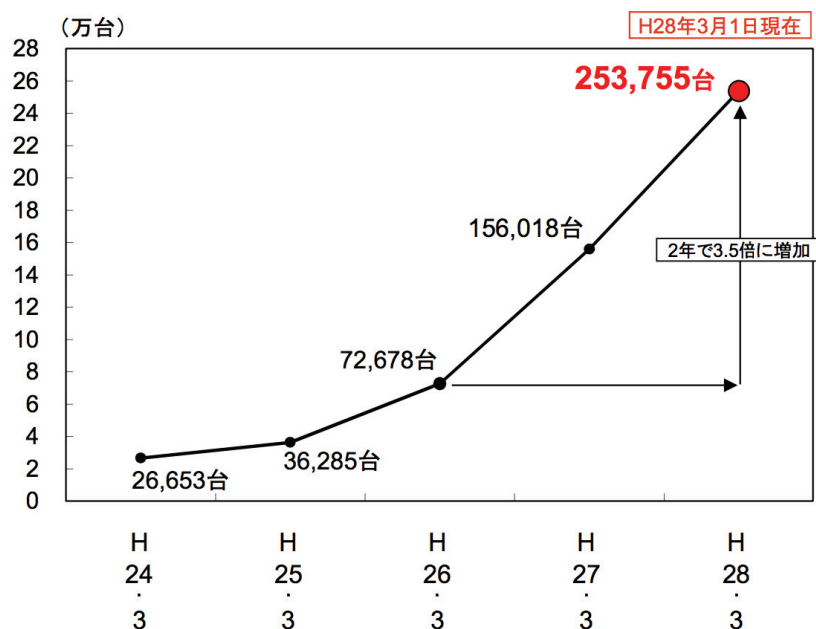


図 1-3 公立学校のタブレット端末配備台数

「教育の情報化ビジョン」策定を受け、実証校として選定された特定の学校では、ICT(Information and Communication Technology)を活用した「新たな学び」を創造するために「フューチャースクール推進事業（総務省）」、「学びのイノベーション事業（文科省）」などの実証研究が実施された。これらの事業は、デジタル教科書・教材の開発、ICTを活用した指導方法の開発、教科指導などにおける ICT 活用の効果・影響の検証を行ったものである。

平成 23 年度から 25 年度までの 3 年間に渡る実証研究の結果が、「学びのイノベーション事業実証研究報告書[27]」にまとめられている。この実証実験では、デジタル教材を Adobe Flash 等により開発していたが、Windows と iOS の学習用端末が混在していたため、OS ごとにそれぞれ別のコンテンツを開発している。実証研究報告書では、「学習者用デジタル教科書・教材等の基本的な考え方」として“多様な情報端末、複数の OS へ対応できること、情報通信技術の進展に応じたものであることが望まれる”と記している。タブレット端末では、Windows、Android、iOS の 3 種が代表的な OS として普及しており、使用環境を OS に縛られない互換性の高い教材開発が今後も求められると予想される。

デジタル教材と発見的学習

学習支援に直接コンピュータを利用する試みは長年続けられているが、大槻[6]によると、直接操作や可視化機能を伴う GUI(Graphical User Interface)技術展開の影響を受けて 1980 年代の半ばから登場した学習方法が、発見型学習である。大槻は、「発見型学習は、人間の視聴覚や感性に訴えて、理解や発見を促すオープンエンド型のパラダイム」であり、「学習者が主体的に新しい知識を獲得し、既得の知識を適用・検証して理解を深めること」が発見的学習の目的である、と述べている。発見的学習によって、体験から獲得された知識は、学習の流れに従った単なるシンボル操作に比べ、はるかに安定的である[26]。

コンピュータの発達・普及に伴い、コンピュータに映像や音声統合されていく度に、これらの機能を用いて学習者の興味を引き出し、発見的学習を行う試みがなされた。メディアの利用は、学習者が興味を持って主体的に学習を開始するきっかけをつくる可能性があり、発見的学習の支援に適している[20]。

デジタル教材の有用性のひとつは、従来の紙メディアの資料や教科書にはない能動的操作である[17]。しかし、現在使用されているデジタル書籍・教材の多くは、従来の教科書に準拠しているため、教科書の内容を電子化し、拡大機能や書き込み機能を付加したものがほとんどである。紙媒体のコンテンツでは表現できない、映像や音声の再生等のデジタルメディアを使用した例はあるものの、3DCG 等従来の教育現場で馴染みの薄い手法を表現方法として取り入れた例は少ない。

従来型の教材やコンピュータでは不可能であった表現方法を、内容に即した手段として幅広く取り入れていくことで、より多様な学習方法を実現できる可能性があると考えられる。

1.1.3 進化教材の不足と必要性

ICT 教育やデジタル教材の普及により、多様な教材コンテンツが発表されているものの、教材数が圧倒的に不足していると言われている分野が、生物分野の進化に関する教材である[15]。

日本では、1993 年に行われた「生物の多様性に関する条約(生物多様性条約)[7]」の調印以降、生物多様性への関心が高まっており、現在に至るまで、生物多様性基本法の制定をはじめとする様々な政策が行われている。教育面においても、平成 21 年 12 月に刷新された高校生物の学習指導要領に、生物基礎の大項目として「生物の多様性と生態系」が取り上げられるなど、生物多様性は重要な単元として位置づけられている[23]。今後の生物多様性の概念と、その重要性の教授手法は重要な研究課題である[15]。

生物多様性は、長期間に渡る生物進化の産物であるため、生物多様性の概念を理解する場合、それが生み出される過程である生物の進化に関する理解が必要条件となる[15]。進化は、生物界全体を統一的に捉え、生物の多様性を理解することに不可欠な概念であり、現在も進行しつつある動的・可変的な自然観を習得する機会を提供するし、科学的リテラシーを育成するなど、その教授価値は高い[23]。

福井ら[23]は、進化の教授価値について、「進化の教授価値は、それが歴史的観点から生物界全体(あるいは生物学という学問全体)を統一的に捉えることができる点にある。そして、その捉え方には、垂直的側面(時間次元における適応的变化)と水平的側面(空間次元における多様化)の両者があり得る。」と述べている。すなわち、時間を経るにしたがって生物が多様性を獲得した歴史的な変遷と、同時代を生きた生物が持つ多様性や斉一性、二つの側面を捉えることで体系的な学習を行うことができる。

現在、進化の理解を促すために、様々な観点からアプローチが行われている。中でも、生物間の系統関係に関する情報を学ぶため、系統樹の学習は必須要素となっている[14]。系統樹とは、生物が共通の祖先から時を経るにしたがって分岐し、種を増やしていったことを、樹木の枝分かれのように描いて示したもの(図 1-4)である。

また、骨格構造の変遷や比較が扱われているケースも多い[5, 12, 13, 22]。これは、「人間を含めた動物の体の特徴、すなわち動物の姿の多様性、進化の一端は骨格を通して観察できる[9]」ことが理由であると考えられる。

しかし、進化は、その現象が長期的な変化の結果生ずるものであること、また、実験的に検証するのが不可能であることなど、教材として取り扱うためには大きな問題を有している[13]。したがって、その取り扱いの難しさゆえに、進化に関する教材開発が急務であるにもかかわらず、教材が圧倒的に不足していると言われている[15]。

また、進化を理解するためには、系統樹や骨格の観察といった複数のカテゴリーから得た知識を複合的に捉える必要があると考えられるが、多くの教材では特定のトピックのみに焦点を当てている傾向が見られる。これは、進化の学習に従来使用されてきた教科書や模型を用いた教材では、「複数のカテゴリーに属する情報を統合して提示する」ことや、「実

験や観察に即した検証が不可能である」点に起因していると考えられる。

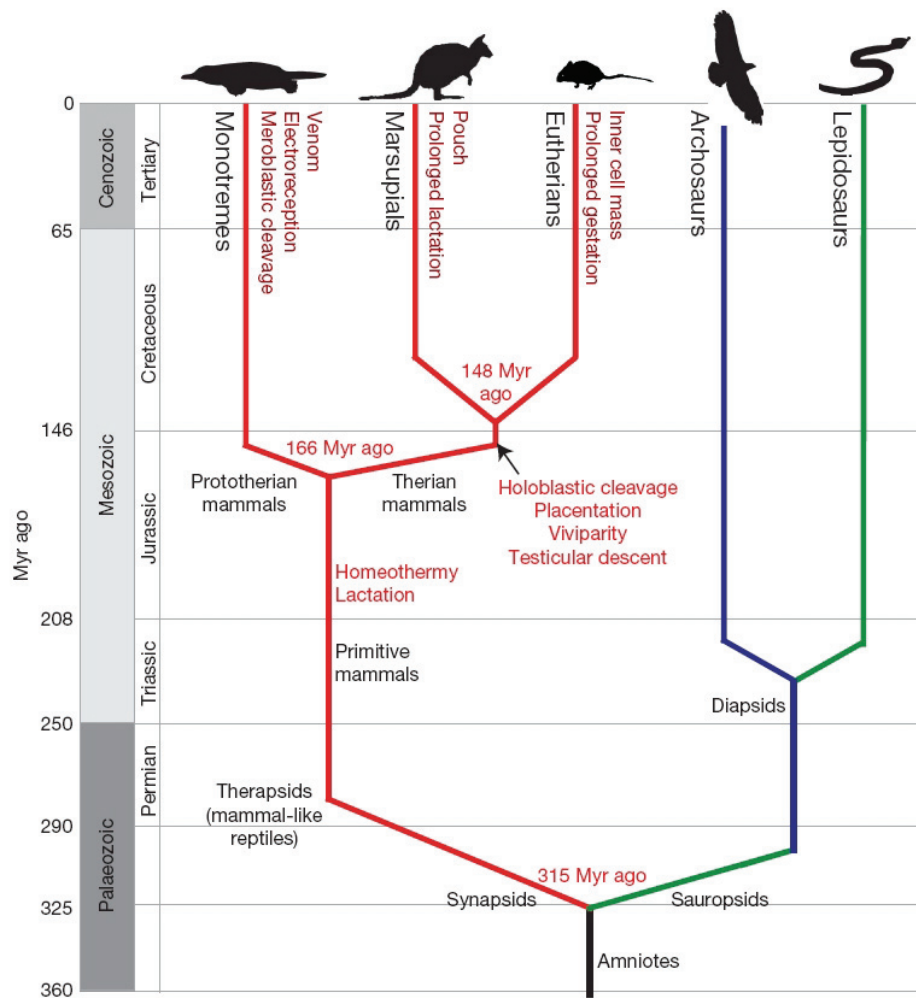


図 1-4 系統樹の例²

² <http://www.nature.com/scitable/topicpage/interpreting-shared-characteristics-the-platypus-genome-44568#>

1.2 本研究のねらい

以上を踏まえ、本研究では、リアルタイム 3DCG 技術の発展に伴う WebGL の一般化や、ICT 教育の普及によるデジタル教材の普及に着目した。

リアルタイム 3DCG は、インタラクティブに様々な方向から観察が可能であり、アニメーション等の表現を付加することができる。リアルタイム CG を使ったデジタル教材は、進化教材が不足する原因となっている表現の困難さを解消できる可能性がある。リアルタイム CG により、従来から進化の教材として使用されている教科書や模型では不可能であった「擬似的な観察」や、「立体構造に対して様々な情報を付与した表現」が可能になる。また、3DCG を Web ブラウザ上で描画する WebGL の技術を使用して教材を開発することで、OS に依存しない互換性の高い教材を開発できる。

さらに、デジタルツールは、学習者が能動的な操作を行うことが可能であり、「学習者が主体的に学習を行い、体験的に知識を理解して獲得する」ことを特徴とする発見的学習に適している。したがって、新たなテクノロジーを利用した従来手法にない表現メディアによる知識の伝達手法を開発することで、学習内容にそぐった、より効果的な知識の獲得を促すことができると考えられる。

したがって、本研究では新たな進化教材として、以下の 3 点に着目したデジタルツールの開発を行う。

- ・ 学習者による能動的操作を基とした発見的学習の支援
- ・ WebGL の利用による互換性の高いデジタル教材の開発
- ・ リアルタイム 3DCG による新たな情報提示の利用

発見的学習に適すると考えられるデジタルツールを用いた新たな進化教材を開発し、リアルタイム 3DCG を用いた新たな情報提示の有用性を確かめることが、本研究のねらいである。

第2章 先行研究・先行事例

2.1 進化教材の開発へ向けた取り組み

前述のとおり、進化の学習は重要な内容であると位置付けられているにも関わらず、教材の圧倒的な不足が指摘されている。進化学習を促すため、これまでも、教材として取り扱う難しさに反して、新たな教材開発へ向けた様々な取り組みがなされている。

系統樹の理解と生物体の観察を用いた授業実践

まず、進化が、長期的な現象であり、実験で検証できないという特性を持つことから、塩野ら[15]はインタラクティブなデジタルツール”TimeTree: The Timescale of Life[21] (図 2-1)”を用いた授業を実践し、有用性を検証している。この研究において、被験者の学生は動物園を巡りながら飼育されている脊椎動物を観察しながら任意の 10 種を選択し、オンラインツールである”TimeTree”を用いて種の分岐年代を検索して、その情報から進化の系統樹を作成した。TimeTree による検索では、2 種類の生物を検索対象として入力することで、両種の学名と分岐年代の要約統計量を知ることができる。学習者は、検索を行いながら種の分岐年代を調べ、得た情報を系統樹にまとめあげることで学習を行う。実習において「検索と系統樹作成」の作業を複数の種で繰り返すことで、進化の学習に必須の要素である系統樹の概念を習得でき、結果的に生物多様性や進化への理解が促進されたとされている。”TimeTree” のシンプルなフィードバックは、学習者の混乱を招きにくいと言われている。塩野らは、ミスコンセプションの多い進化の分野では積極的にアクティブラーニングを実施すべきであると述べている。

この研究で用いた”Time Tree”のフィードバックは、文字やグラフといった 2 次元情報にとどまっている。したがって、進化の理解に重要であると考えられる、生物体や骨格の視覚的な観察を行うことができないという欠点を有しており、塩野らはその点を動物園における観察という形で補完していたと考えられる。生物体や骨格は立体的な構造を持つものであり、これらを多角的に観察するためには、実物の観察や標本、模型等を使用した学習が別途必要となる。また、進化という一つの単位に対して、動物園等の外部施設を使った実習を行うことは難しく[12]、実現には困難が伴うと予想される。

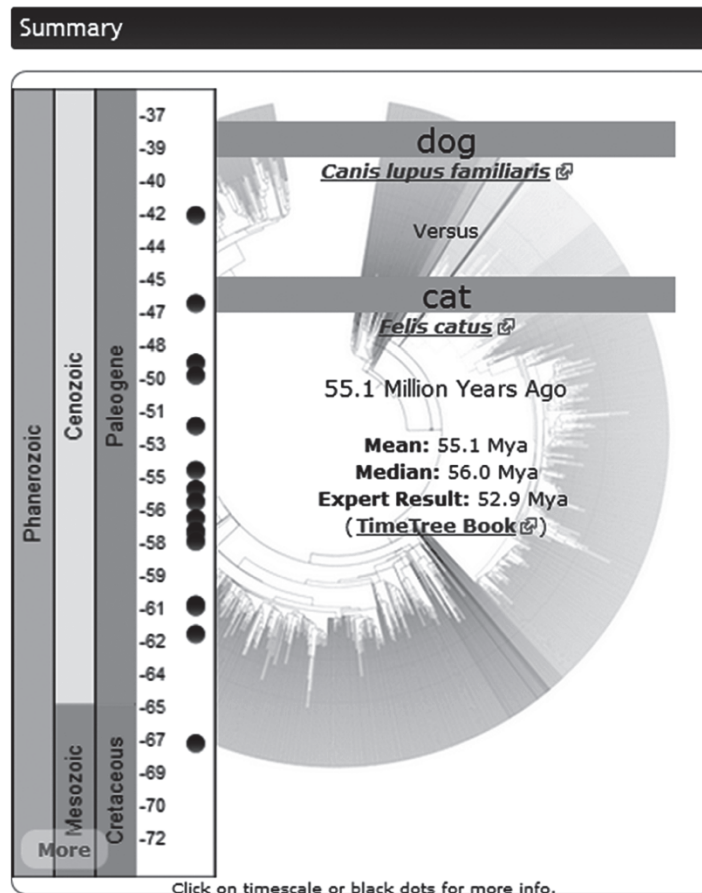


図 2-1 TimeTree: The Timescale of Life の検索結果例

実物の標本と 3D 画像の利用

標本を積極的に授業へ取り入れた例としては、太田[5]の研究が挙げられる。太田は、従来の授業の問題点として、

- ・ 生物の進化を理解させるための化学的根拠である「化石標本や骨格標本」の希少性が高く、授業に活用される機会が少ないこと
- ・ 教科書などに掲載された「平面的な写真資料や模式化した図表等」を用いた間接的な学習にとどまっていたこと

を挙げている。そこで、この研究では、医科大学や博物館との連携を積極的に行い、本物や実物に近い立体的な標本を活用した授業を展開した。標本を活用した観察・実験の結果から、生徒らは実感の伴った論理的な考察が可能となったことが示唆されている。

しかし、実物の標本を生徒一人ひとりに対して確保して観察・実験を行わせることは困難である。そこで、太田は授業のために借用した標本を 3D スキャナーでデータ化し、3D プリンターによるレプリカの出力や、バーチャルスライド用の 3DCG 生成を行っている。

これにより、何度でも立体的な構造の閲覧が可能となり、具体的な比較や考察を伴う学習活動を導入することができた、と述べられている。

2.2 形態比較を目的とした 3DCG の活用

実物に代わって 3DCG を使用する利点は、立体情報の伝達が容易であるのみでなく、デジタルデータでなければ不可能な表現ができる点にも挙げられる。筆者らが行った研究 [29] では、2 種類の骨格の比較を、CG モデルのモーフィング処理による変形表現を利用して行うことで、閲覧者に骨格における新たな発見を促した。

CG モデルのモーフィング処理とは、3DCG モデルの頂点情報を利用した変形アニメーションの一種である(図 2-2)。CG アニメーションでは、キャラクターの表情作成に利用されることの多い手法である。このアニメーションには、モーフターゲットと呼ばれる頂点と面の数や構成が一致した複数の 3DCG モデルと、実際に変形する CG モデルであるベースメッシュの用意が必要となる。ベースメッシュに対してモーフィング処理を行うことで、ベースメッシュが持つデータに、モーフターゲットごとの形状情報が追加される。ターゲットに応じた変形量(Morph target influence 値)をベースモデルに与えてモデルの頂点座標を制御することで、図 2-2 のように変形を行うことが可能となる。変形アニメーションは、それぞれの対応する頂点の座標値を線形補完することで得られる。

この研究は、生物体の骨格構造の共通点と差異に着目し、それらを 3DCG の変形表現によって読み取りやすくしたものである。生物の骨格構造は種によって異なり、それぞれが複雑な形態を有しているものの、基本的な構造が似通っている。

ウマとヒトの前肢骨格の場合、その形態は大きくことなるものの、同一部位として分類された骨の対応関係をみると図 2-3 のようになっている。同じ色の骨が同一部位として対応関係にある。また、ヒトの骨格のうち灰色に着色された骨は、ウマには存在しない部位である。このように、いわゆる中指のみで体を支えており、指を構成する骨の数も少ないことが読み取れる。

博物館において多くの骨格標本は、骨格標本を並列して展示されている。しかし、閲覧者が生物分野における専門知識を持っていない場合、骨格標本のみの展示から同一部位とされる骨を認識することは極めて難しいと考えられる。

そこで、この研究では、同一部位の比較に、同一部位の対応関係に従ったリアルタイム CG のモーフィング処理による変形アニメーションを利用した。アニメーションを眺めているだけで、閲覧者が変形前と変形後の 2 者を比較することができる点から、同一部位の認識を計り、形態差異に対する新たな気づきや発見を促した。例えば、ウマとヒトの全身骨格を比較した場合、3DCG モデルの変形アニメーションは、図 2-4 のように再生される。

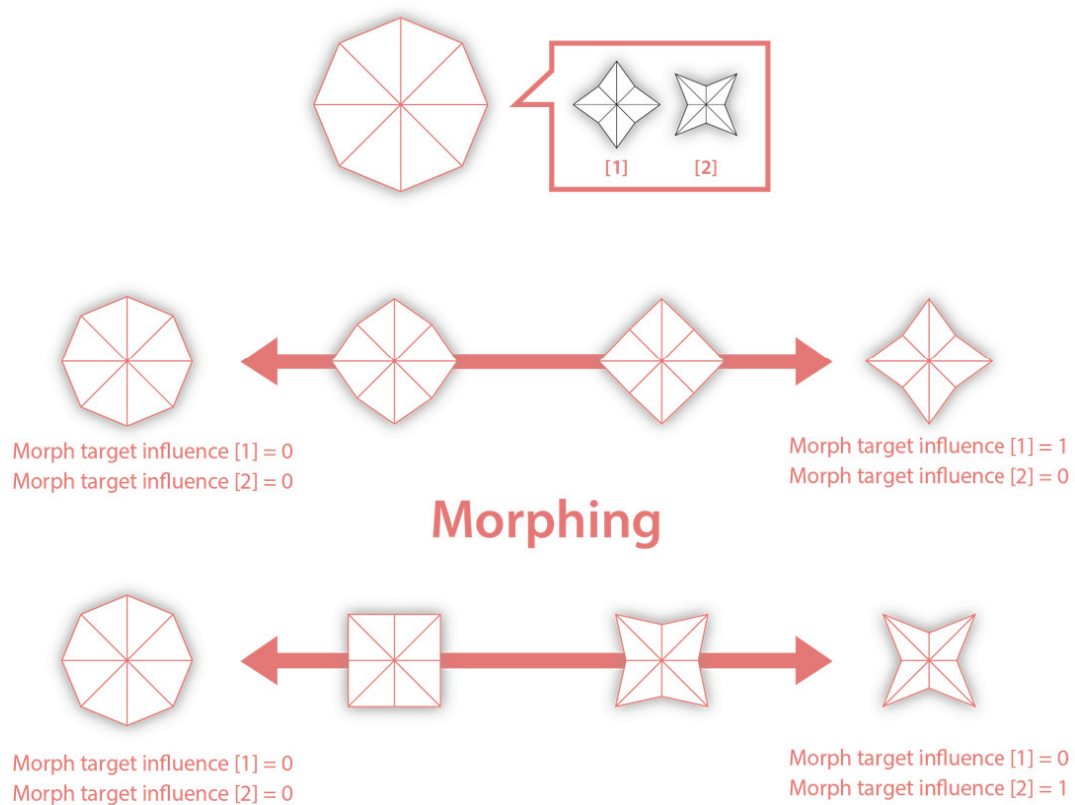
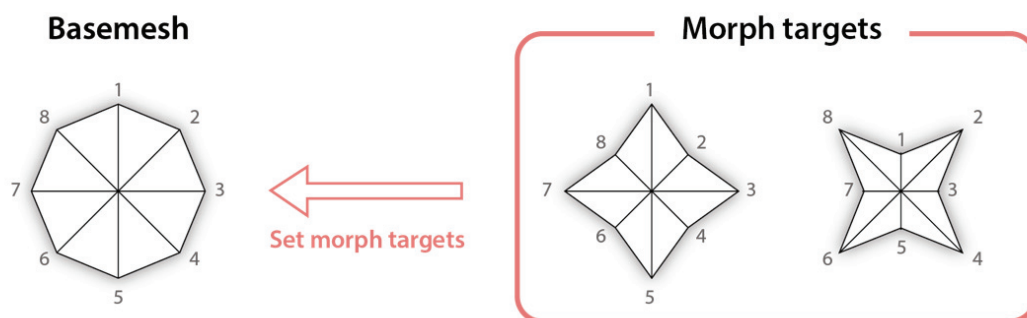


図 2-2 3DCG モデルのモーフィング処理の概略図

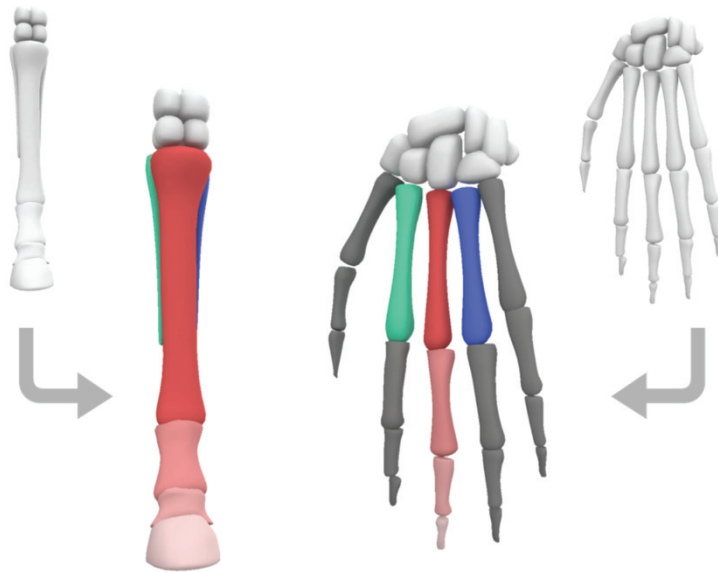


図 2-3 ウマとヒトの前肢における骨格の対応関係

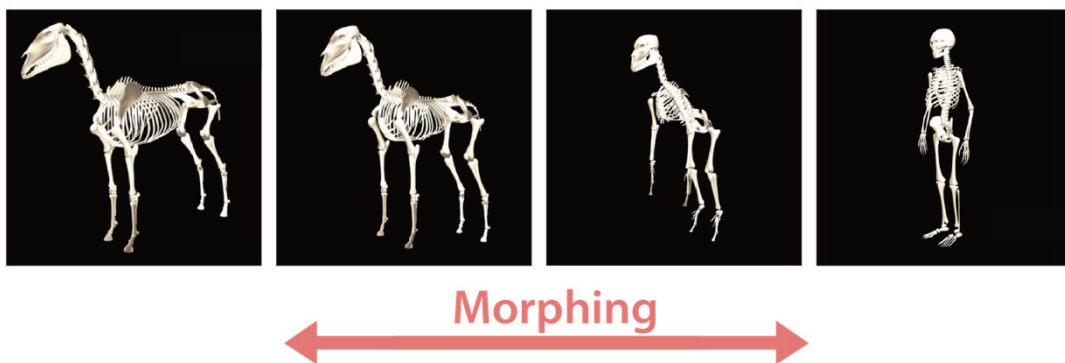


図 2-4 変形アニメーションの様子

この手法で制作したコンテンツを骨格モデルの閲覧や、変形アニメーションの再生が可能な Web ページとして実装し、骨学に対する専門知識を持たない 12 名に、自由に閲覧してもらったところ、比較した 2 種類の骨格における共通点や差異について、全員が新たな気づきや発見を行うことができた。発見の内容は、「ウマには中指以外の指が無い」、「哺乳類の骨格の構成は意外と似通っている」といった、専門知識無しに 2 種類の骨格を見比べただけでは、自ら把握することが難しいものであった。従来の骨格標本展示では不可能な”自発的な発見”が、このような骨格 3DCG モデルの変形を利用することで可能になったと言える。

上記で述べた変形アニメーションを使った情報提示は、異種間の骨格比較における発見的学习に適していると考えられる。骨格構造の変化は、生物の進化の過程に密接に関わったものであるため、この情報提示の方法を用いれば、学習者が自ら「骨格構造の比較」から「進化の過程」に対する発見を行うことができる可能性がある。

しかし、この変形アニメーションの提示のみでは、「比較している 2 種が進化前後の関係である(例：ウマがヒトに進化した)」という誤解を招きかねないという問題点があり、閲覧者が間違った知識を”発見”してしまう問題があると示唆されている。

また、評価実験で行われた「閲覧から生物に対する興味・関心の向上」に関する評価は 5 段階評価の平均値が 4.1 となり、同時に調査された「閲覧の楽しさ」「比較の容易さ」といった項目に比べ最も低い値となっている。骨格モデルや変形アニメーションの提示のみでは、閲覧者の「生物に対する興味・関心」を伸ばすことは難しく、その点においては、コンテンツに対するより詳細な情報の提示が有効である可能性が考えられる。

2.3 WebGL を利用したデジタル教材

3DCG の閲覧には、従来専用のソフトウェアが必要とされていたが、2011 年に Web ブラウザ上で 3DCG を描画可能な規格として WebGL が発表されたことにより、3DCG を用いたデジタルツールの一般化が容易になった。これまでは、端末ごとに環境設定が必要だった 3DCG による表現が、Web ページにアクセスするだけで閲覧可能になったことから、デジタルツールの可能性は立体にまで広がったと言える。

3 次元的な構造に対する教材として、WebGL を利用した教材開発を行った研究が山森ら[30]によってなされている。山森らは、HTML5 および WebGL を用いて、天体シミュレーションによる小学生理科補助教材(図 2・5)の開発と、その有効性の評価を行った。

この研究では、小学生の理科の単元である「太陽と月」が題材として取り上げられている。この単元は、教科書のみでは 3 次元空間における複数の天体の動きがイメージしにくいこと、実際に観察ができないことから、学習が難しいとされている。この教材には、WebGL の特徴を活かし、天体の動きや月の見え方をインタラクティブなリアルタイム CG で表示する機能が実装されている。

教材に対するアンケートでは、「わかりやすい」、「(天体の相対的な関係が)実感できる」といった好評価が得られた。生徒の視点からは「楽しく学習できるため、学力の向上につ

ながる」といった感想が、教師の視点からは「実際に使って教えたい」などの要望が数多く挙げられた。特に好評価だった教材は「月の見え方のリアルシミュレーション」であったが、これは、天体の相対位置と月の見え方の関係が直感的に把握できるためと考えられる。

また、吉富[31]も、WebGL を用いた天体学習用の 3D 教材を開発している。吉富は、天体の学習が難しい理由として、「天体の動きを知るためには、3次元空間における物体の運動を2次元平面に投影した教科書の図を見ながら、頭の中で3次元的な天体の動きとして復元し、理解する必要があること」として、3次元的な動きの認識・把握を助ける目的で教材の開発を行った。

この3D教材は、タブレット端末で表示可能な教材であることを条件として制作されたため、プラグインの導入が必要のない WebGL が導入された。生徒一人一人にタブレット端末を配ることで、個々の理解度に応じて自由に端末を操作することが想定されている。

吉富は、WebGL のメリットとして、「インタラクティブなコンテンツでありながら、端末の中にローカルファイルとして保存しておくことで、インターネット接続ができない環境でも同じように表示することが可能である」ことを挙げている。通常の CGI を用いて作成したコンテンツでも、インタラクティブな Web ページを作成することが可能であるが、CGI はサーバ上で動作するためオフラインでは使用できない。その反面、HTML5 と JavaScript を用いたコンテンツの場合、動作はクライアント側で行われるため、多数の端末を用いた際にインターネット環境が乱れても対応することが可能となる。

以上の事例が示す WebGL の特徴は、進化教材の開発に対しても有効であると考えられる。

進化の教材として頻繁に用いられている生物の骨格は、立体構造が複雑であり、従来は実物や模型が無ければ多角的な観察が難しかった。しかし、そのような題材でも、WebGL を利用することで、Web 上で簡単にコンテンツを閲覧できるため、実物の模型なしに構造の観察を行うことが可能となる。また、Web 上への公開が容易にできるため、コンテンツの普及促進についても大きな利点を有している。

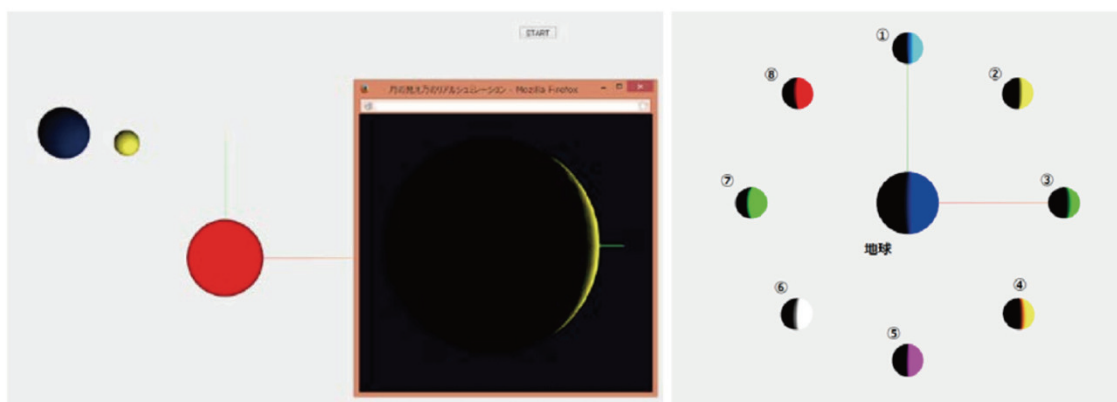


図 2-5 天体シミュレーション教材の実行画面

2.4 タッチデバイスの教育への活用

ICT 教育の普及に伴い、教育現場へのタブレット端末をはじめとするタッチデバイスの配備台数は飛躍的に向上している。

前述した吉富の研究[31]でもタブレット端末が用いられている。

中井ら[19]は、タブレット端末を利用したデジタル教材の開発を目的として、3D コンテンツを作成する際の提示手法を考案し、有用性を検証している。

中井らは、2次元コンテンツと比べ、3次元コンテンツには形状を理解するための情報量に大きな違いがあり、直感的な理解を促す教材として3次元コンテンツの表現力は大きな力になると述べている。タブレット端末を利用したアプリケーションはコンピュータでの操作とは異なるため、直感的な操作や手軽に利用できる利点が期待される。

しかしながら、コンテンツそのものが3次元であっても、ディスプレイ上に実際に描画されるコンテンツは2次元であるため、直感的な3次元コンテンツの理解を促すためには3次元空間の比較的単純な操作方法が必要となる。タブレット端末には加速度センサー等さまざまな操作方法が考えられるが、この研究では、「提示された画面を見ながら操作することや一般的な学習状況での利用を考慮」し、タッチパネルでのマルチタッチ操作が考案された。3次元空間の操作は、一度にタッチする位置を同時に3点までに限定して実装されている。

実装された操作は以下のように行われる(図 2-6)。

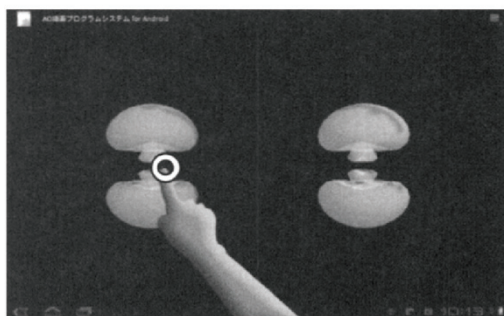
- ・ 3D モデルの選択処理：立体モデルを1点でタッチ
- ・ 3次元空間のカメラ位置操作
 - ・ 拡大・縮小(カメラを注視点に近づける)：2点でピンチ
 - ・ 回転(3次元空間そのものを外から回転させるような動き)：2点タッチした状態でもう1点をスライド
 - ・ 平行移動：3点を同時にスライド

実装後に生徒・児童を対象に行われた実証実験では、操作習熟度にばらつきはあるものの、殆どの生徒は比較的簡単に操作方法を覚えることができた。3D空間を使った教材により、空間内のオブジェクトの位置関係・大きさの比較や把握において習得に十分な効果が見られたと述べられている。

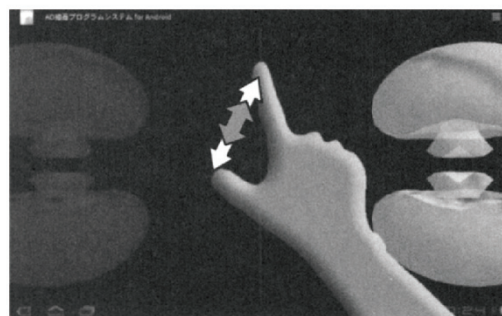
また、一部のコンテンツでは、3Dモデルの回転軸を固定して、1点をスライドすることで回転させていたものの、その点が各自の操作において問題となるケースは少なかった。

しかしながら、3次元空間内のカメラ操作をマルチタッチで行った際、画面上に描画されたオブジェクトの位置を見て、直接オブジェクトをタッチしてしまうといった事例も見られ、インターフェース上での単純操作が必ずしも直感的な操作に結びつくわけではないことが確認された。

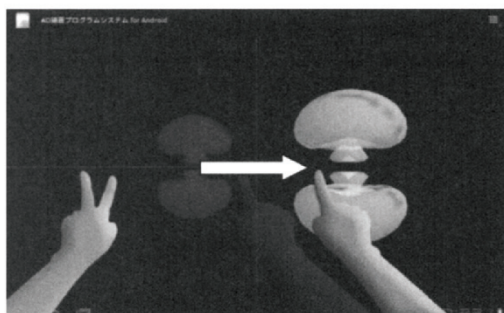
タッチデバイスを用いた3次元空間特有の提示方法や操作方法は、コンテンツの内容に応じて検討していく必要があることが示唆されている。



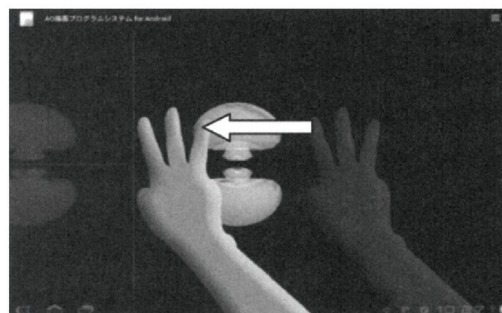
(a) 選択



(b) 拡大・縮小



(c) 回転



(d) 平行移動

図 2-6 中井らによるタッチ操作

第3章 提案手法

3.1 提案手法の概要

近年、デジタルツールの普及を背景に、学習者が能動的な操作を行うことのできる教材が Web 上で公開されている。このようなツールを使って学習することで、学習者は自らの体験から知識を獲得する発見的学习によって、安定的な知識を獲得できると考えられる[20]。これらのコンテンツは、公開が Web 上で行われることから、閲覧や普及の促進が容易であるという利点を持つ。

デジタル教材の導入が進む一方で、高校・大学の生物分野において、「進化」に関する教材の不足が指摘されている[15]。この原因は、進化が、その現象が長期的な変化の結果生ずるものであること、実験的に検証するのが不可能であることが挙げられる。また、進化を理解するためには、系統樹や骨格の観察といった複数のカテゴリから得た知識を複合的に捉える必要があると考えられるが、多くの教材では特定のトピックのみに焦点を当てている傾向が見られる。

そこで、本研究では、進化への自発的な発見を促す新たな教材として、3DCG モデルの変形アニメーションと系統樹型の UI を用いた、リアルタイム CG 骨格比較ツール(以下、比較ツール)を提案する。学習者は比較ツールを使用する際に、系統樹型 UI から比較対象の 2 種を選択することで、閲覧する変形アニメーションを選択し、立体構造の比較を行うことができる。このツールにより、学習者は「系統樹の学習」、「骨格構造の観察」、「骨格の比較による自発的な知識の獲得」という 3 つの観点から進化について学習を行うことが可能になる。複数の観点から進化を学ぶことで、学習者は複合的な視点から、能動的に進化を捉えることができると予想される。このツールの利用から、学習者に進化への理解を促すし、リアルタイム 3DCG を用いた新たな情報提示の有用性を確かめることが、本研究の目的である。

中でも、「骨格構造の観察」、「骨格の比較による自発的な知識の獲得」に関しては、CG モデルのモーフィング処理による変形表現を利用することで、複雑な骨格構造の比較が可能となり、閲覧者自らによる自発的な知識の獲得が可能になると考えられる。また、CG モデルへテキストを付加し、閲覧の目的に応じたカメラの制御を行うことで、立体構造に付随する情報提示を行い、従来型の教材とは異なった手法で発見的学习の支援を行う。

ICT 教育を行う際のデジタル教材としての汎用性を高めるために、Web 上で閲覧・公開が可能な教材とし、特に教育の現場で普及が進むタブレット端末による使用を想定し、開発を行う。

3.2 提案の目的

デジタル教材の普及を背景に様々な ICT 教材の開発が行われているが、現在も不足が指摘されているものが進化に関する教材である。進化に関する学習の重要性に反し、進化を

扱った教材は少ない。この原因は、進化が、その現象が長期的な変化の結果生ずるものであること、実験的に検証するのが不可能であることが挙げられる。また、進化を理解するためには、系統樹や骨格の観察といった複数のカテゴリから得た知識を複合的に捉える必要があると考えられている。進化の教授価値は、生物学全体を「垂直的側面(時間次元における適応的变化)」と「水平的側面(空間次元における多様化)」の両者の観点から捉えることができる点にあると言われている[23]。そのため、進化の理解を促すためには、生物間の系統関係に関する情報を持つ系統樹の学習や、骨格の観察を通じた種の多様性や変遷の学習が行われている。しかし、多くの教材では、進化という現象特有の表現の困難さゆえに、特定のトピックのみに焦点を当てている傾向が見られる。

しかし、近年進むデジタルツールの普及を背景に、従来と異なる形で情報を伝達できる教材の制作が可能となり、学習者が能動的な操作を行うことのできる教材が WebGL によって制作されるようになった。このようなツールを使って学習することで、立体構造に対する擬似的な観察が可能になるほか、学習者は体験から自発的に知識を発見する「発見的学習」によって、より安定的な知識を獲得できると考えられる。合わせて、これらのコンテンツはブラウザ上で動作する Web アプリケーションとして公開することができるため、OS に縛られない汎用性の高い教材を実現することが可能であり、閲覧や普及の促進が容易であるという利点を持つ。

そこで、本研究では、進化への理解を促す教材として、系統樹と骨格を題材とした Web ベースの CG 骨格比較ツールを提案する。提案するツールは、「系統樹の学習」、「骨格構造の観察」、「骨格の比較による自発的な知識の獲得」という3つの観点から進化を学習するための教材である。複数の観点から進化を学ぶことで、学習者は複合的な視点から、能動的に進化を捉えることができると予想される。このツールの利用から、学習者に進化への理解を促し、リアルタイム 3DCG を使った情報提示の発見的学習への有用性を検証することが、本研究の目的である。

3.3 進化学習を目的とした設計

3.3.1 従来型の進化教材：前肢の骨格比較

現在使用されている進化に関する教材の中で、骨格の比較を行っているトピックをベースとして、開発を進めることとする。

そこで、中学/高校生物の教科書(2016 年度現在)において必ず掲載されている「前肢の骨格比較(図 3-1)」を題材として取り上げる。このトピックでは、生物の相同器官である前肢/前腕/ヒレに着目し、それらが進化の過程で生活様式に沿って形態を変えていったことを扱っている。相同器官の骨格を比較することで、異なった種における骨格の多様性と類似性を読み取ることができる。

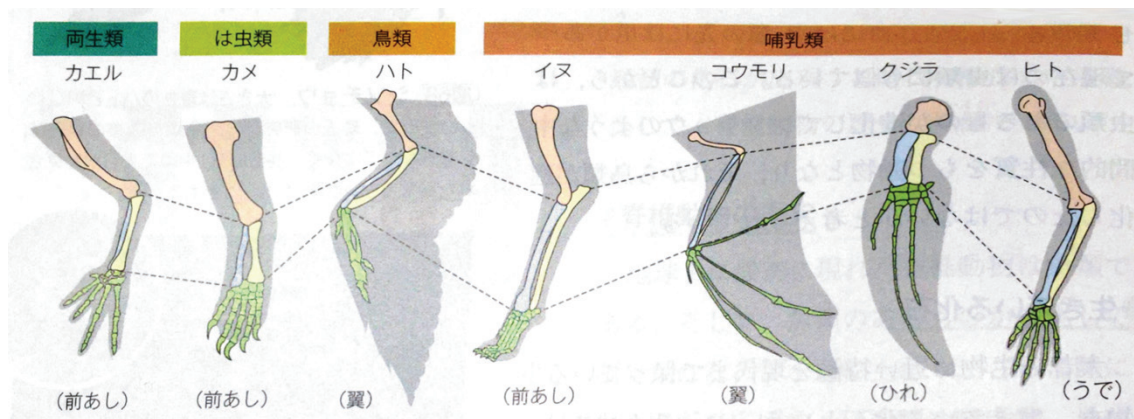


図 3-1 前肢の骨格比較[18]

3.3.2 3DCG の変形表現

進化の学習において、骨格の比較は頻繁に行われる内容のひとつである。従来型の進化教材で骨格の比較は、ある一方向から構造を捉えた2次元的なイラストによって行われているが、生物の骨格構造は本来、非常に複雑な立体構造を有している。したがって、立体構造の観察に適するリアルタイム 3DCG を用いることで、自由度の高い学習を行うことができると考えられる。

開発のベースとしている「前肢の骨格比較」では、構造に共通点と差異の両者を含む骨格を比較している。筆者らの行った先行研究により、そのような立体構造の比較を行う際には、モーフィング処理を利用した変形表現が有効であることが分かっている。変形アニメーションの閲覧により2種の骨格構造を比較することで、専門知識を持たない閲覧者でも、特別な解説を必要とせずに複雑な立体構造を有する骨格の共通点や差を読み取ることが可能である。したがって、比較の際に 3DCG の変形表現を使用することで、学習者が自ら新たな知識を獲得する「発見的学習」の支援が可能になると考えられる。

そこで、本ツールでも、骨格構造の比較を変形アニメーションによって行うこととする。学習者は比較対象となる2種類の生物を選択し、変形アニメーションを再生することで生物の骨格構造の比較を行うこととする。変形アニメーションは、骨格の対応関係にしたがって設定される。

また、従来教材である「前肢の骨格比較」で生物の形態を比較する際に描かれることの

多い、生物体のシルエットを同時に描画し、骨格モデルの変形アニメーションの再生に合わせて同様の変形を付加することとした。

3.3.3 系統樹を使用した比較対象の選択

筆者らの行った先行研究では、CG モデルの変形アニメーションによる単純な形態の比較において適切な情報提示がなされない場合、「ウマがヒトに進化した」といったような間違った発見を促す危険性があり、進化への誤解を生む可能性が示唆された。

そこで、本研究では、進化の学習に必須であると言われる要素のひとつである系統樹に着目した。提案手法では、比較対象の2種を選択するために設置が必要と考えられるモデル選択パネルを、系統樹を模した UI(以下、系統樹型 UI)として教材に設置する。系統樹を使って学習者自らにモデルの選択を行わせることで、進化への誤解を防ぎかつ系統樹への理解を深める一助にできると予想した。

学習者は、比較ツールを使用する際に、まず系統樹型 UI(図 3-2(a))で比較したい種を2つ選択する。図 3-2 でイラストとして描かれたノードは種を、エッジは進化の過程を表している。比較ツールでは、学習者の選択した種に応じて、変形アニメーションの閲覧が可能となる。この時、閲覧者の選択が図 3-2(b)のように系統樹において「祖先・子孫」の関係にある場合、変形アニメーションには「進化の過程」に着目して再生される。一方で、選択された2種が図 3-2(c)のように「祖先・子孫」の関係にない場合、変形アニメーションは単なる「骨格の比較」として再生される。

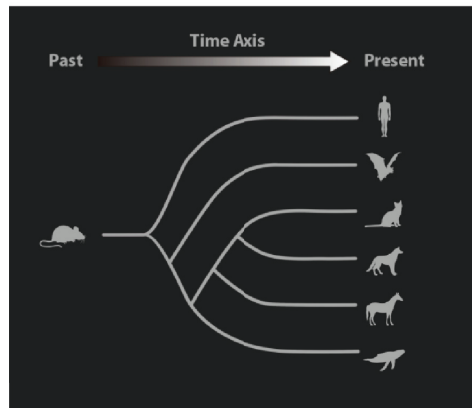
学習者は、このような UI を使用して比較する2種を自ら選択し、変形アニメーションによる骨格比較を閲覧することで、図 3-2(b)の場合は進化の「垂直的側面」を、図 3-2(c)の場合は「水平的側面」を、骨格という観点から発見できると予想される。

また、系統樹によって骨格の CG モデルと変形アニメーション以外の情報を追加することで、先行研究に比べ、学習がより「生物への興味・関心の向上」につながると予想される。

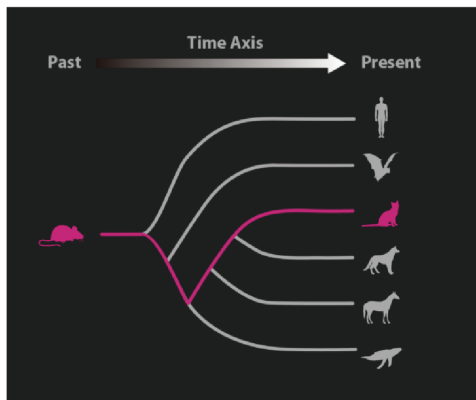
3.4 想定する使用環境

本研究では、ICT 教育の普及により著しく普及が進むタブレット端末を使用環境として想定して開発を行う。タッチデバイスであるタブレット端末は、指で直接タッチして操作を行うことから、直感的な操作が可能となっている。実物の標本を観察する際と近い感覚で、能動的な CG モデルの観察できると考えられる。また、タブレット端末で3D空間を使用した教材閲覧することで、空間内のオブジェクトの位置関係・大きさの比較や把握が容易になると予想される。

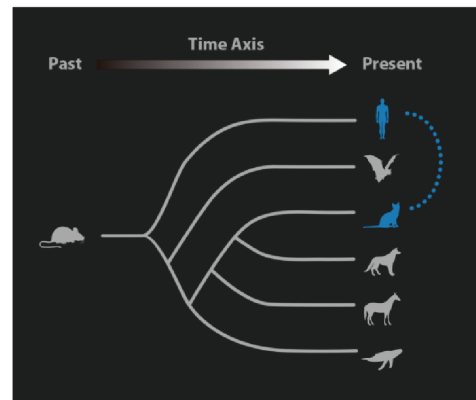
ただし、提案する比較ツールは WebGL を用いて制作されるため、WebGL に対応したブラウザを有する端末であれば、PC やスマートフォンでも同様に閲覧することが可能である。



(a)



(b)



(c)

图 3-2 系統樹型 UI

第4章 リアルタイム骨格比較ツールの開発

4.1 開発環境

リアルタイム骨格比較ツールの開発に用いたハードウェア、ソフトウェア、言語およびライブラリは表 4-1 の通りである。

表 4.1 開発環境

名称		主な用途
ハードウェア	iPad mini	コンテンツの閲覧
ソフトウェア	Autodesk Maya 2015	3DCG モデルの作成
	Adobe Photoshop CS6	UI およびテクスチャの作成
	Adobe Illustrator CS6	UI およびテクスチャの作成
	Safari (バージョン 601.1)	<iPad 端末> コンテンツの閲覧
言語	Python (2.7.10)	3DCG モデルのファイル形式変換
	HTML (HTML5)	コンテンツの実装
	JavaScript (1.7)	コンテンツの実装
ライブラリ	Three.js (r66)	Web ブラウザ上での 3DCG 描画
	jQuery (1.12.3)	コンテンツの実装
	jQuery UI (1.12.1)	コンテンツの実装

ハードウェアには、iOS のタブレット端末である iPad mini を選択している。iPad mini は、iOS タブレットのシリーズの中でも比較的サイズが小さく、価格も安価であり、教材としても活用されているモデルである。解像度は 1024×768px、メモリは 512MB となっている。2010 年にはじめて iOS タブレットとして発売された iPad 1 を除けば、iOS のタブレット端末に搭載されたメモリの下限は 512MB である。リアルタイム 3DCG コンテンツは、その特性上大量のメモリリソースを必要とするが、iPad mini で問題なく動作するコンテンツを開発することで、ほとんど全ての iOS タブレット端末で安定した動作を見込むことができる。

なお、3DCG モデルの制作に使用した Maya は、モーフィング処理の設定が可能であるために制作ソフトウェアとして選択している。Maya において、モーフィング処理はブレンドシェイプデフォーマ機能として提供されている。

また、three.js は WebGL の機能をプログラミングしやすくまとめた JavaScript ライブラリである。Python スクリプトは、three.js ライブラリに同梱されている 3D モデルのファイル形式変換スクリプト”convert_obj_three.py”を実行するために用いている。

4.2 題材の選定

本研究では、教科書に掲載されている教材「前肢の骨格比較」をベースとして開発を行う。制作する 3DCG モデルは、生物の左前肢の骨格および生物体のシルエットである。

題材とする生物の選定にあたり、従来の教材「前肢の骨格比較」に取り上げられた生物から比較対象として取り上げられる頻度の高い、哺乳類を対象とした。中でも、相同器官の形態に特徴の見られる生物を選定することとする。

まず、教科書に掲載されている哺乳類から「ヒト、コウモリ、イヌ、ネコ、ウマ、クジラ」の 6 種を選択した。これらは、いずれも現生生物である。

提案手法では、現生生物の比較と合わせて系統樹や長時間にわたる生物の進化を取り上げる。そこで、現生する哺乳類の共通の祖先 [28] として、現在発見されている最古の哺乳類 [3] である「アデロバシレウス」を選択した。この生物は、中生代三畳紀後期の 2 億 2,500 万年前に生息していたと推定されている。推測される体長は 10cm から 15cm ほどで、現在のトガリネズミのような姿をしていたと考えられている。

4.3 3DCG モデル変形表現の実現

提案手法でモデリングが必要となる CG オブジェクトは、骨格モデルおよびシルエットに該当する表皮（以下、このモデルをスキンモデルとする）である。この 2 種類のモデルには、モーフィング処理による変形表現を設定できる形式で、制作を行う必要がある。

変形可能な 3DCG モデルを制作するために行った手順を以下に示す(図 4-1)。

まず、解剖学等の資料を参考にしながら、ベースモデルとなる CG モデルを 3DCG 制作ソフトウェアでモデリングする。制作する 1 つめのモデルに当たるベースモデルの制作は、基本的に通常のモデリングと同様に進めるが、後に設定する他のモデルと大きく形状が異なる部分には、あらかじめ変形を考慮して分割数を増やしておく。1 つめのモデルが完成したら、テクスチャを設定するために、UV マッピングを行う。

2 つめ以降のモデルは、モーフターゲットとなるモデルである。モーフィング処理による変形アニメーションの設定には、頂点や辺といったオブジェクトの構成を一致させる必要があるため、これらのモデルはベースモデルの頂点を移動させて作成する。

必要なオブジェクト全ての準備が整ったら、モーフィング処理を設定する。制作に使用した Maya において、この機能はブレンドシェイプデフォーマ機能として提供されている。この作業によって、構成が一致した、形態の異なるモデルを出力することが可能となる。

骨格モデルの場合は、この段階で複数のパーツを一つのモデルとして結合する。

上記で準備したベースオブジェクトに、変形量を設定する Morph target influence 値を変更しながら、頂点座標、テクスチャ座標、法線ベクトル、フェース構成を含む OBJ フォーマットでそれぞれのモデルを出力する。

モデルの準備が終了した後に、OBJ フォーマットのモデルを、Web ブラウザで読み込み可能なファイル形式に変換する。

出力した OBJ モデルを、three.js ライブラリに同梱されている Python スクリプト”convert_obj_three.py”を用いて、JSON 形式で記述された JavaScript ファイル(JS ファイル)へ統合し、ファイルの形式をブラウザで読み込める形式へと変換する。

出力される JS ファイルには、3DCG 制作ソフトウェア上でモーフィング処理を設定した時のように、モーフトーゲットの頂点情報が morphTargets 配列として記録されている。ベースモデル以外のモデルがもつ立体形状は、この配列に格納される。

上記で作成した JS ファイルを、WebGL を利用して Web ブラウザで読み込むことで、CG のリアルタイムレンダリングおよびインタラクティブなモデルの変形が可能となる。

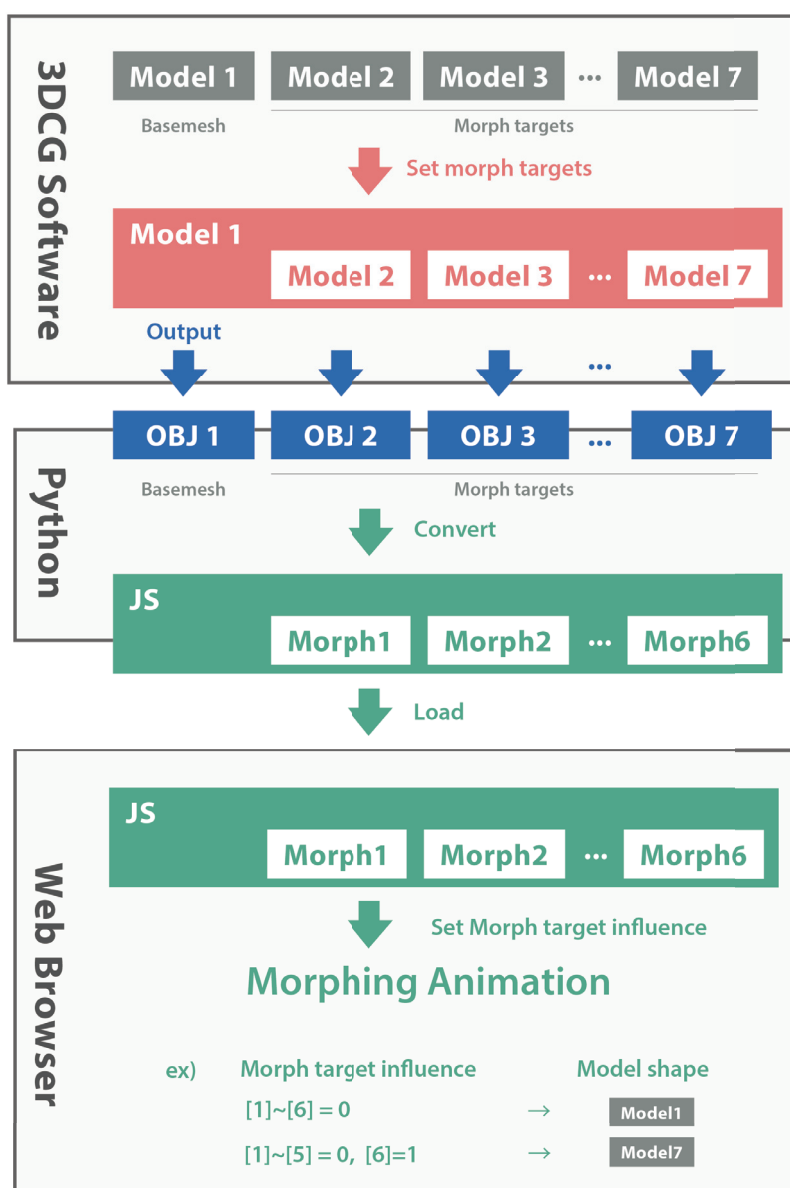


図 4-1 モデルファイルの制作

図 4-2 は、制作した CG モデルの一覧である。iPad 等のタブレット端末では、PC と異なり 3DCG 閲覧に十分なメモリや GPU が搭載されていないことが多いため、これらモデルはなるべく不要な頂点を持たないように制作されている。



図 4-2 制作した CG モデル

最終的に出力された JS ファイルは、
骨格モデル：頂点数 約 3300、ファイルサイズ 約 800KB
スキンモデル：頂点数 約 900、ファイルサイズ 約 200KB
となっている。

3DCG モデルの変形アニメーションは、モーフィング処理の変形量を、ベースモデルに対して動的に設定することで実現される。変形量を制御する Morph target influence 値を JavaScript によって変更すると、その値に対応する配列 morphTargets に格納された頂点情報にしたがってベースモデルの頂点位置が線形補完される。

この Morph target influence 値を微量に変更しながら連続してレンダリングを行ってディスプレイを随時更新することで、変形アニメーションを提示することができる。

実際の比較ツールにおいて、モデルの変形アニメーションは、図 4-3 のように再生される。

また、学習者の入力に従って Morph target influence 値を動的に設定することで、インタラクティブに CG モデルを変形することが可能となる。

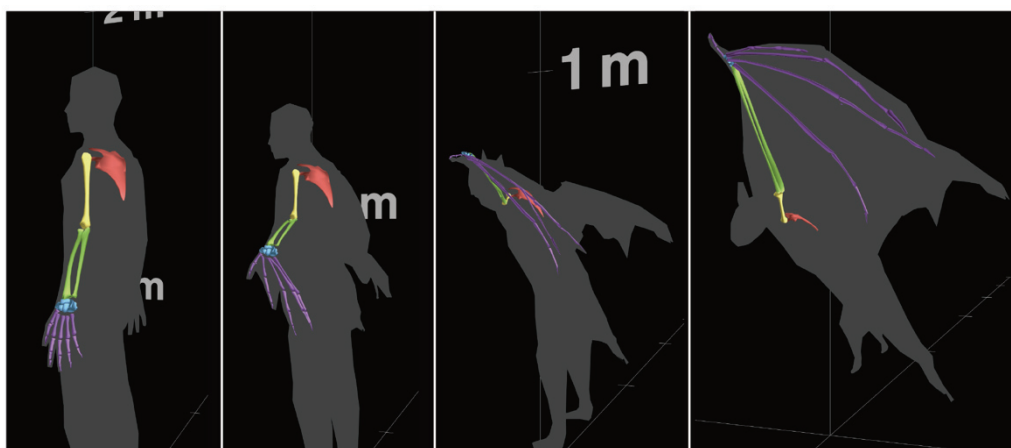


図 4-3 変形アニメーションの様子(ヒトとコウモリ)

哺乳類の骨格構造には類似性がみられるものの、生物の種によって、骨格を構成する骨の数は異なっている場合も多い。例えば、前述したとおりヒトとウマを比較すると、両者では指の数が異なっている。ウマは中指にあたる第3指のみで立っている状態であり、進化の過程で、その他の指は第2指と第4指の中手骨を除き退化消失している。

モーフィング処理の設定には、CGモデルの立体構造が一致している必要があるため、上記のように、骨の数が比較する2種で異なっている場合でも、頂点数等を一致させる必要が生じる。

そこで、骨格構造に差がある場合の変形アニメーションは、図4-4のように再生されるようにモデルを制作した。

ヒトとウマの第4指に着目すると、ヒトは第4指を有しているが、ウマの第4指は中手骨を除いて進化の過程で退化消失したと考えられている。図4-4で青色に描画されている部分は、本来ウマが持たない骨であるが、その部分のCGモデルを縮小し、ウマの第4指中手骨のモデル内部(図4-4右端の赤丸部分)に置いている。このようにモデルを制作しておくことにより、図4-4の左側の状態から変形アニメーションを再生した場合、「ウマが持たない部分の骨が小さくなって消えた」ように見える。

上記の方法で、他の種を比較した際の組み合わせにも考慮してモデル制作を行うことで、CGモデルの立体構造を維持した状態で、多様な骨格構造の比較アニメーションを実現することが可能となる。

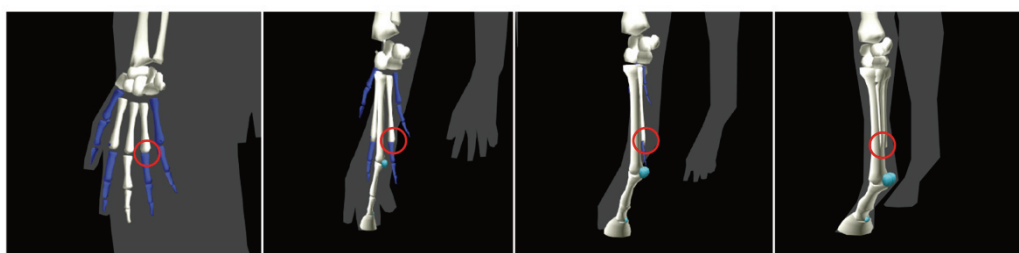


図 4-4 骨格構造に差がある場合の変形表現

4.4 レンダリング順の制御

本ツールで3次元空間上に描画する内容は、「骨格、生物体のシルエット、スケール」の3つである。骨格とシルエットは教材の題材として必須であるほか、実空間のように比較対象となる物体が存在しない仮想空間においてオブジェクトのサイズを把握するためには、目安となるスケールが必要となる。本ツールでも、サイズ感の把握を目的として、1m 間隔の目盛りを持つスケールを描画し、縦方向のみ「1m、2m、…」とサイズを記載したラベルを設置することとした。

従来の紙メディアに描かれた「前肢の骨格比較」のイラストでは、手前に骨格が描かれ、

その背面にシルエットが描かれているものが一般的である。しかし、3次元空間では前面・背面という概念が基本的には存在せず、モデルやその頂点・面の位置は3次元空間における座標として定義されている。したがって、2次元のイラストにおいて骨格とシルエットとして描かれていたものを3次元空間で再現しようとした場合、実際の生物のように骨格とシルエットに該当する表皮をCGモデルとして制作し、仮想空間に配置することとなる。そのままレンダリングした場合、スキンモデルに骨格モデルが覆われる形となってしまう、レンダリング画像にはスキンモデルのみが描画されることになってしまう。

これは閲覧の補助として配置するスケールについても同様であり、同一の3次元空間に優先度の異なるモデルを配置する場合に起こり得る問題である。

そこで、本ツールでは、レンダリング順の制御を行うことで、どのような角度からモデルを閲覧しても骨格モデルが手前に表示されるようにした。骨格モデル、スキンモデル、スケールを図4-5のように重畳して表示する。

優先度の低い情報から先にレンダリングし、優先度の高い情報を後から重ねていくことで、3次元空間における描画情報にプライオリティを持たせることで、必要な情報を読み取りやすくするねらいがある。これにより、3D空間におけるモデルの位置に関わらず、常に骨格モデルが手前に表示される。

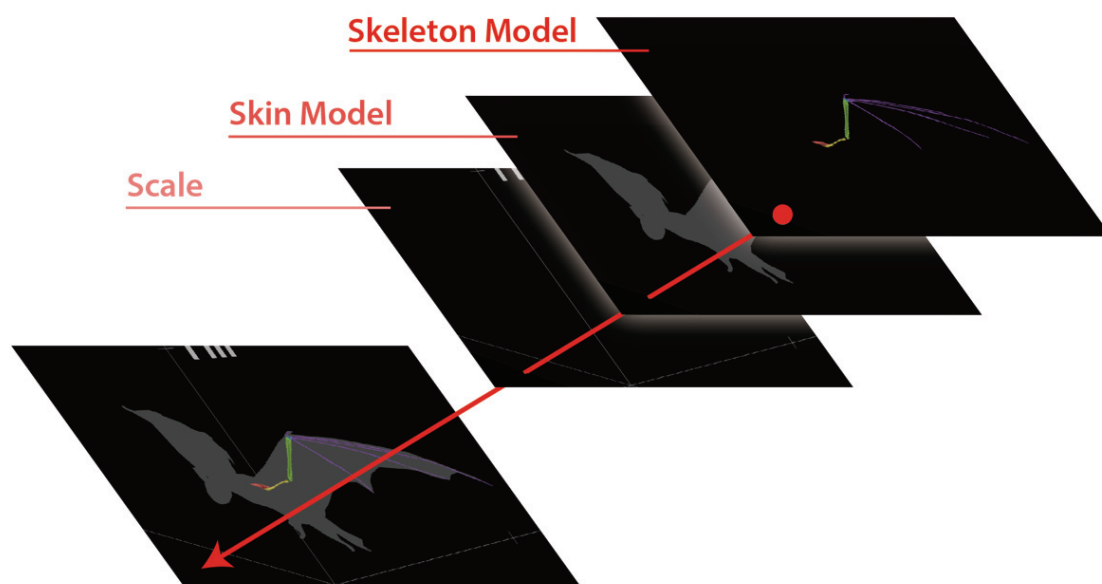


図 4-5 レンダリング順の制御による重畳表示

4.5 画面構成と機能

4.5.1 画面構成

本研究で開発した比較ツールは、「系統樹の学習」、「骨格構造の観察」、「骨格の比較による自発的な知識の獲得」という3つの観点から進化を学習するための教材である。これらの観点からの学習を実現するため、3DCGの変形アニメーションと系統樹型UI、それに付随する機能をひとつの画面内に構成した(図4-6)。筆者らの先行研究[29]に比べ、今回のコンテンツでは比較対象となる3Dモデルの数が増加している。そこで、教材自体をひとつのWebページ上にまとめ、より快適に比較・観察を行えるように配慮した。これにより、ページ遷移によるデータ読み込みの待ち時間を省き、学習の分断を防ぐねらいがある。また、学習者が自らの観察から知識を発見することを支援するため、教材全体としてCGモデルと系統樹を除く補足情報は設置していない。また、この画面構成は後述する操作性に関する予備評価実験を踏まえて改良されたものである。

画面は、3Dビュー、操作パネル、系統樹型UIの3つに大別される。

まず、3Dビューには、骨格モデル、スキンモデル、スケールがリアルタイムに表示される。3Dビュー上でタッチ操作を行うことで、カメラの位置を変更することが可能となる。3次元空間の広がりを感じられるように、端末の画面サイズに関わらずブラウザ画面全体に表示されるように実装されている。また、操作パネルや系統樹型UIの背面にも描画を行うことで、画面サイズが比較的小さいタブレット端末でも、3DCGの観察をより大きな領域で行うことができる。

次に、操作パネルからは、3DCGモデルの変形量やテクスチャ、カメラモードを制御することができる。操作パネルのUIは、直感的な操作ができるように、なるべく文字による表記を減らし、グラフィカルな表現を用いて作成した。

系統樹型UIは、比較対象のモデルを選択するために使用する。系統樹には選択肢である哺乳類のアイコンが描かれており、それらをタップすることで比較対象を選ぶことができる。

4.5.2 3Dビューの操作方法

タッチ操作による3Dビューの操作方法是、中井らの研究[19]を参考に設定を行った。

ただし、本ツールでは、3Dモデルの選択処理を行う場面がないため、先行研究で選択処理に割り当てられていた「1点での操作」を他の処理に割り当て、より単純化を計る。

実装された3次元空間のカメラ操作は以下のように行うこととする。

- ・拡大・縮小：2点でピンチ
- ・回転：1点をスライド
- ・平行移動：3点を同時にスライド

操作パネル

3D モデル / カメラモードの操作

- 変形量のリセットボタン
- 変形量の調整スライダ
- メッセージ
- 変形アニメーションの再生ボタン
- ビューの自動回転ボタン
- テクスチャの切り替えボタン
- カメラモードの切り替えスイッチ

モデルの選択ボタン (アイコン)

系統樹型 UI

比較対象の選択

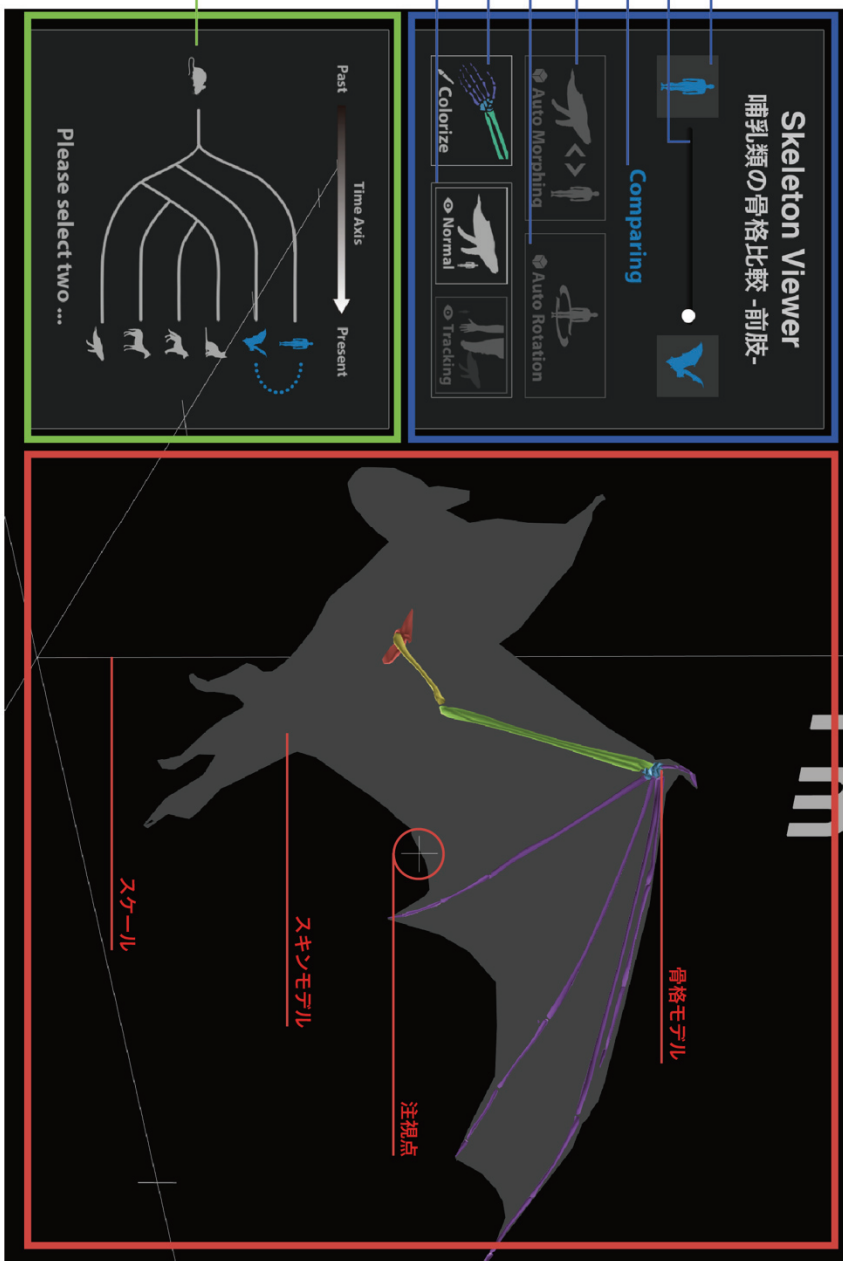


図 4-6 画面構成

なお、3D ビュー以外の2次元で表現された部分(操作パネルおよび系統樹型 UI)は、通常の2次元コンテンツと同様の操作方法で動作するように実装した。タップされた位置を検出して、操作が3D ビューに対して行われたものか否かを判断することで、「1点での操作」を複数の処理に割り当てている。

また、後述するカメラトラッキングモードでは、3D ビューのカメラ位置をプログラム側で自動的に制御しているため、

- ・ モデルの回転：1点をスライド

のみが、タッチ操作として受け付けられる仕組みとした。

実装には、Three.js の OrbitControls クラスを用いている。

タッチデバイス以外の PC 端末等でツールを閲覧した際には、マウス入力により、以下のように3D ビューの操作を行うことができる。

- ・ 拡大・縮小：中ドラッグ
- ・ 回転：左ドラッグ
- ・ 平行移動：右ドラッグ

4.5.3 系統樹型 UI による比較対象の選択

学習者は、画面左下に設置された系統樹型 UI のアイコンをタップすることで、比較対象となるモデルを選択する。2種の生物が選択された段階でモデルが読み込まれ、3D ビューに3D モデルが読み込まれる。系統樹型 UI の操作に応じて、操作パネルのアイコンおよびメッセージも更新される仕組みとなっている(図4-7)。

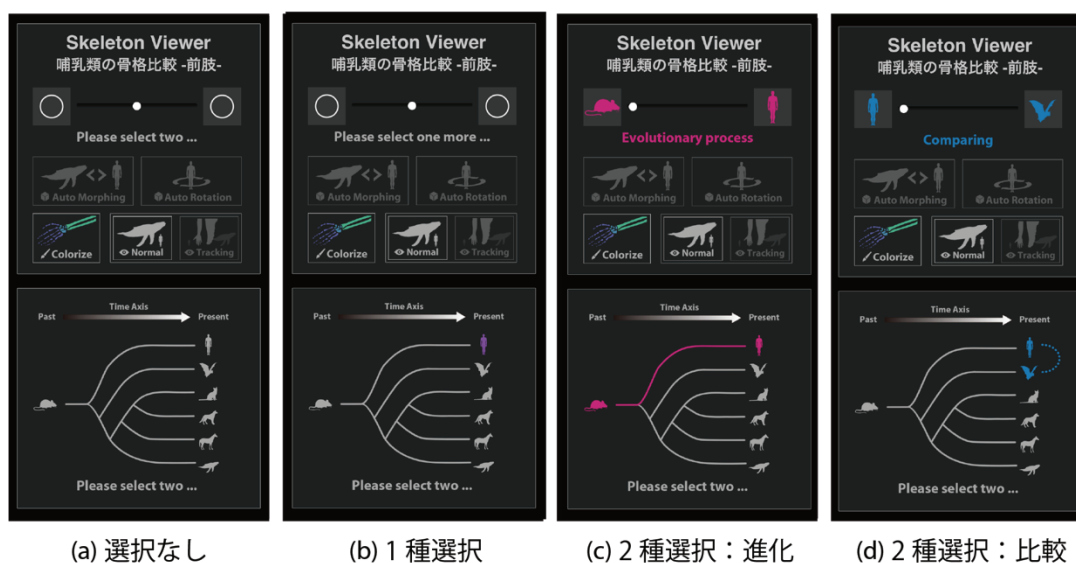


図 4-7 系統樹型 UI の選択に応じた動作

比較対象が1つも選択されていない初期状態が図 4-7(a)にあたる。選択されたアイコンは(b)のように紫色になる。この状態からもう1種を選択することで、「進化の過程」または単なる「生物の比較」として変形アニメーションを閲覧することが可能となる。すでに選択されているアイコンをもう一度タップした場合は、その種の選択が解除される。また、2種が選択された状態で3種目を選択した場合は、先に選択されていた2種両方の選択が解除される。

学習者が、選択した2種の関係性を考えながら閲覧できるように、2種が「祖先・子孫」の関係にある場合は赤系統、祖先-子孫の関係にない場合には青系統の色を使っている。また、2種の関係性をより明確にするために、(c)では2種をつなぐ系統樹のエッジに色付けを、(d)では2種をつなぐ点線を表示した。

4.5.4 変形アニメーションの再生

CG モデルの変形アニメーションの制御は、操作パネルから行うものとする。変形アニメーションに関わる操作は以下の3箇所から行うことができる。

- ・ 変形量のリセットボタン：アイコンをタップすることで、3DCG モデルの変形量がリセットされ、アイコンに表示された種の骨格を閲覧できる。
- ・ 変形量の調整スライダ：学習者が、任意に変形量を調整できる。変形アニメーションの再生ボタンが ON になっていた場合は、アニメーションの変形段階に応じて自動的に動く。
- ・ 変形アニメーションの再生ボタン：ON にすると、変形アニメーションが「種 A → 種 B → 種 A → ...」のようにループ実行される。OFF にするとアニメーションが停止する。

4.5.5 閲覧支援機能

4.5.5.1 カメラトラッキング

今回選択した題材の哺乳類は、アデロバシレウス(体長約 15cm)とクジラ(体長約 15m)といったように体長に大きな差がある。サイズ差の大きい CG モデルで変形アニメーションを再生した場合、再生の最中にモデルの大きさが大きく変わっていくこととなる。その際、モデルがカメラの枠からはみ出てしまったり、モデルが小さくなりすぎて見えなくなってしまうたりすることが予想される。

そこで、サイズの異なるモデルを比較する際の閲覧支援機能として、3次元空間におけるカメラの動作モードを2種類用意した(図 4-8)。

ひとつは、通常のカメラと同様の動作をする「ノーマルモード(図 4-8(a))」である。

ノーマルモードでは、カメラを学習者が自由に動かすことができたため、自由度の高い観察や、特定の場所をより詳細に見ることができる。

また、3次元空間では、配置されたオブジェクト以外にサイズ感を把握するための比較対象が存在しない。スケールの設置はしているが、比較する2種の大きさを比べる際には、カメラの位置を固定した状態で変形アニメーションを閲覧する必要があると考えられる。

もうひとつが、モデルの大きさに応じて自動的にカメラの位置を制御する「トラッキングモード(図 4-8(b))」である。

こちらのモードでは、変形アニメーションの段階に沿ってカメラの位置を動かすことで、モデルが常に画面内に収まるようになっている。特定の頂点に着目して変形アニメーションを再生した場合、その頂点が描画される画面位置の移動が、ノーマルモードに比べ大幅に少なくなる。したがって、モデルのサイズ差に関わらず形状の比較が行いやすくなると予想される。

トラッキングモードは、大きさのそれほど変わらない種を比較した場合でも有効であると考えられる。

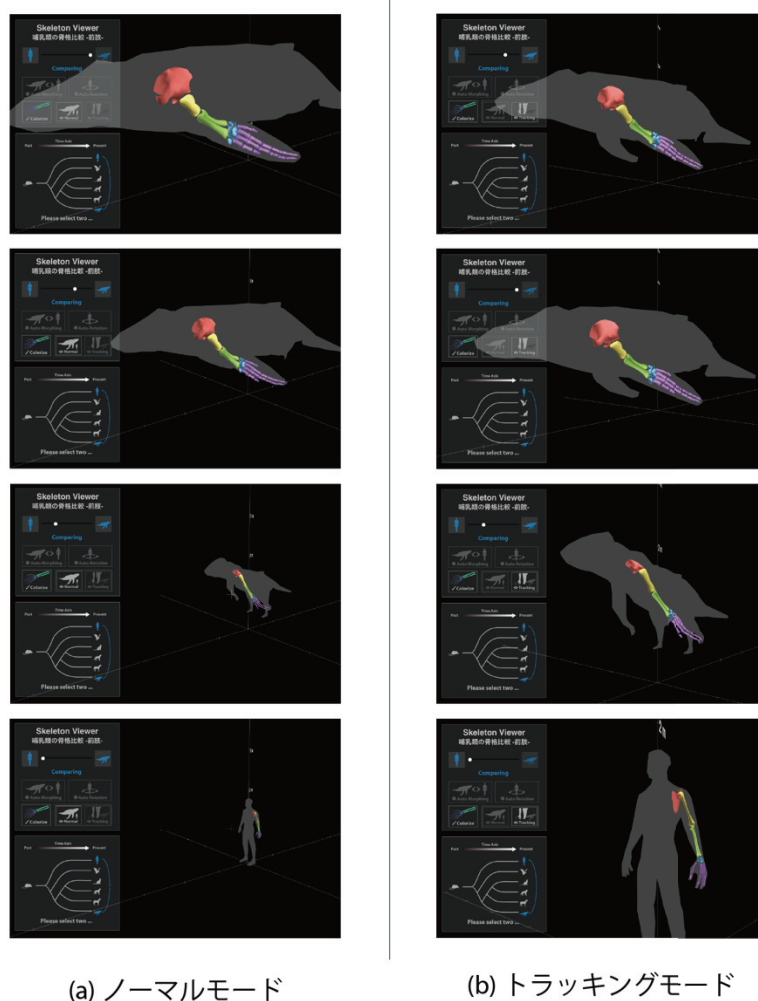


図 4-8 カメラモード

4.5.5.2 テクスチャによる配色方法の切り替え

骨格構造の共通点や差を明確にするために、テクスチャの切り替え機能を実装した。テクスチャの配色方法は、下記の2種類が用意されている(図 4-9)。テクスチャの切り替えボタンを押すたびに、2種類の配色方法が切り替わるようになっている。

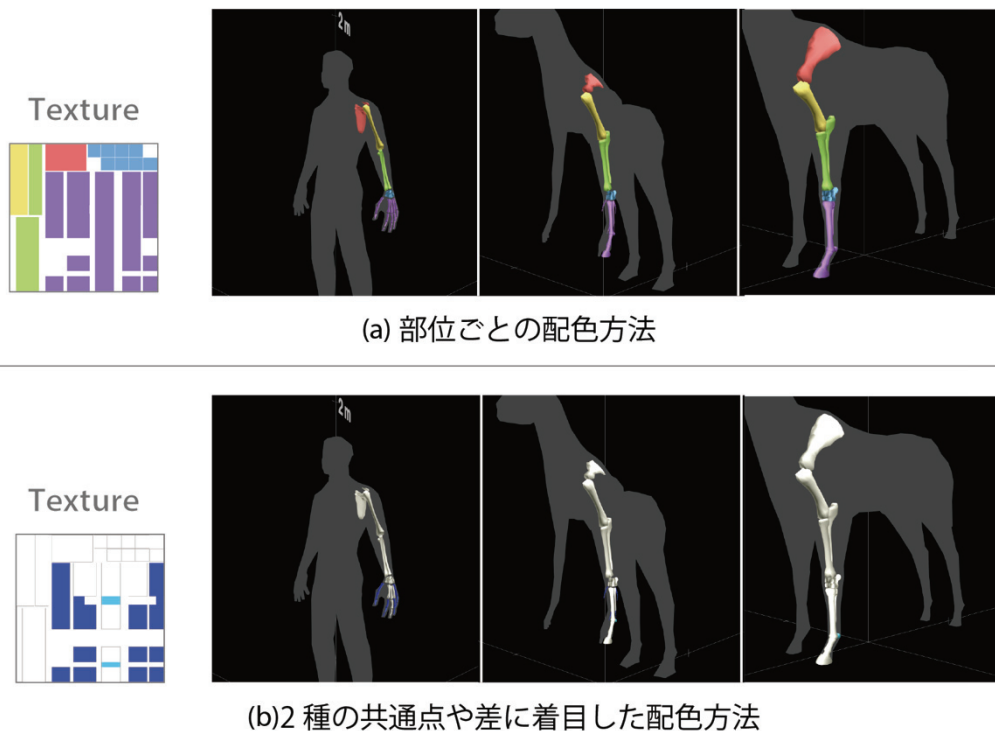


図 4-9 2 種類の配色方法

1 つは、従来の教材にしたがって、骨格の部位ごとに配色を行う方法である(図 4-9(a))。ヒトで説明した場合、骨格を「肩甲骨、前腕、上腕、手根骨、手」に分け、それぞれに別の配色を割り当てている。

もう一つは、選択している 2 種の共通点や差に着目して配色を行う方法である。

ヒトとウマの比較を例に挙げると、図 4-9(b)の白い部分は「ヒトとウマどちらにもある骨」、青く配色されている部分は「ヒトにはあるがウマには無い骨」、水色に配色されている部分は「ウマにはあるがヒトには無い骨」となる。配色には、系統樹型 UI や操作パネルのメッセージと同様に、選択した 2 種の関係性を考えながら閲覧できるよう、2 種が「祖先・子孫」の関係にある場合は赤系統、祖先・子孫の関係にない場合には青系統の色を使っている。

このテクスチャを適用して変形アニメーションを再生することで、配色された骨が現れたり消えたりすることが容易に分かるため、骨格の構造に関する直感的な形態差異の把握

に役立つと考えられる。

この2種の共通点や差に着目した配色方法では、比較対象として選択された2種類に応じてテクスチャを切り替える必要がある。本ツールでは、比較対象に7種のモデルを選択しているため、 $7C_2 = 21$ 通りのテクスチャが用意されている。

4.6 操作性に関する予備評価実験

開発中に、制作したプロトタイプを大学院生4人に見せ、操作性に関する意見を自由に述べてもらった。

その結果、タッチデバイスを用いた3次元空間の操作自体は問題なく習得できるとの意見が得られた。タッチデバイスによる3次元空間の操作はまだ一般的でなく、3DCGを使ったゲーム等のアプリケーションでも操作の軸が制限されていることが多い。しかし、操作方法が比較的単純であり、タッチデバイスで2次元コンテンツを扱う動作に似通っている点からも、基本的な操作の習得に問題はないとみられる。

その一方で、カメラの回転やズームの起点となる注視点がわかりにくく、操作にとまどいが生じてしまうことがわかった。

特に回転操作について、画面が意図しない方向へ動作することから、3D酔いの懸念があることが示唆された。多くの3Dアプリケーションでは、3次元空間の操作における困難さを排除するため、前述のとおり操作が制限されている。カメラ自体の高さが変更不可となっている場合や、カメラを回転させる代わりに3Dモデルを回転させるような操作が設定されている。予備実験でも、「カメラを回転する動作」を「3Dモデルを回転する動作」とであると認識してしまったがために、思った通りにコンテンツを動かすことができなかったと考えられる。

4.7 変更点および追加機能

予備実験を踏まえ、カメラの注視点を、タブレット端末画面の中心から、コンテンツ右側の3Dビューのみが表示されている部分の中心に設定しなおした。また、この注視点はズームの起点となることから、注視点の位置を十字で描画することとした。図4-6は、改良後の状態である。

注視点の描画には、3D酔いを防ぐ狙いもある。注視点の描画位置を変更することで、回転操作の軸が明確になり、ズームの中心点も判別しやすくなったと考えられる。

第5章 評価実験

提案手法として開発した、リアルタイム CG 骨格比較ツールについて、評価実験を行った。

5.1 実験の目的

評価実験の目的は、以下の3つである。

1) 提案手法により発見的学習の支援が可能か

提案手法であるデジタル教材「リアルタイム CG 骨格比較ツール」の閲覧が、学習者が自ら能動的に知識を発見し獲得する「発見的学習」につながるのかを、実験によって検証する。また、CG モデルと系統樹を除く補足情報を含まない提案手法から、学習者はどのような内容の発見を行うことができるのかを調査する。

2) 提案手法と従来手法による学習の比較

3DCG やその変形アニメーションを用いた新たな教材の表現方法を用いた学習と、従来の紙メディアの教材を用いた学習を比較する。同じ情報量を持つ教材の表現方法が、「デジタル教材特有の新たな表現」と「従来型の表現」とで異なっている場合、その違いが学習者への知識の伝達にどの程度影響を与えるのかを探ることが目的の一つである。

3) 提案手法の印象・操作性への評価

提案手法の発展可能性や改善点を探るために、印象・操作性に関する評価を行う。

5.2 実験方法

5.2.1 被験者

実験では、比較ツールを用いた学習により、学習者が新たな知識を自らの観察によって発見することができるかを確かめる点にある。したがって、大学以上の教育機関において、生物学や解剖学に関する専門知識を学んだ経験のない人を実験の対象とした。

実験に参加した被験者は、21 歳から 27 歳までの男性 11 名、女性 9 名の計 20 名である。高校生物を履修していた被験者が 12 名、高校生物を履修しておらず生物分野の学習は中学理科までとなっている被験者が 8 名となっている。

5.2.2 実験手法

実験は、被験者を A と B の 2 グループに分けて行った。実験の手順を以下に示す。

1. 事前知識に関する事前アンケート
(Aのみ) 提案手法の操作説明・操作練習
2. (A)提案手法 / (B)従来手法 による教材の閲覧 (7 分間)
3. 教材内容に関するテスト
(Bのみ) 提案手法の操作説明・操作練習
4. 提案手法の自由閲覧 および発見的学習に関する自由記述アンケート
5. 提案手法の印象・操作性に関するアンケート・感想

手順 1 の事前アンケートでは、主に被験者の事前知識についてアンケートを行っている。被験者が生物分野に対する専門知識を有していないことを確かめた。

その後、グループ A に対しては、iPad mini を用いた提案手法の操作説明および操作練習を行った。グループ B に対する提案手法の説明・練習は、提案手法を実験に使用する手順 4 の前で行われる。この時使用される練習用の Web ページは、提案手法に含まれる CG モデル・アイコン・系統樹を異なるデータに置き換えたページである。練習の際に読み込まれる題材には、生物に関する知識の習得に差が生じないように、球体のような抽象的な形状のモデルを使って行った。実験中に提案手法の練習を行うタイミングが異なっているが、手順 2 の学習と手順 3 のテストに影響は無いと考えられる。

手順 2 の、時間制限を設けた教材の閲覧では、提案手法による教材を閲覧するグループ(A)と、従来手法による教材を閲覧するグループ(B)で学習に使用する教材が異なっている。従来手法である紙メディアの教材は、提案手法から読み取ることのできる「前肢の骨格構造、生物体の大きさ、系統樹」の情報と同等のものを構成した。なかでも、「前肢の骨格構造」を伝えるための紙教材は、教科書に掲載されている「前肢の骨格比較」にならって構成を行った。教材の内容に差が生じないように、従来手法の構成には、提案手法で使用している CG モデルを基に制作した画像を用いた。

制限時間は、提案手法と従来手法の内容を閲覧するために十分な時間であると考えられる 7 分間を設定している。

グループによって異なる教材を閲覧した後、手順 3 では被験者に対して教材に関するテストを出題した。テストの出題内容には、本来の進化学習で暗記する必要のない知識も含まれているが、提案手法と従来手法の表現の違いが、知識の定着にどの程度影響を与えるのかを確かめるために実施している。

テストの後に、グループ B にのみ、提案手法の操作説明と練習を行う。

ここからの手順はグループに関わらず同様である。

手順 4 では、被験者に提案手法を時間制限せずに自由に閲覧してもらい、合わせて「教材の閲覧によって新たに知ったこと、推察したこと、疑問に思ったこと」を問う

自由記述のアンケートを実施した。この自由記述アンケートは、提案手法の、発見的学習に対する適性を確かめるために行っている。

手順5では、最後に提案手法に関する印象・操作性や感想を問うアンケートを実施した。アンケート項目は一部を除き5段階評価となっている。

5.3 実験結果

5.3.1 教材内容に関するテスト

手順3で行った教材内容に関するテスト結果について述べる。提案手法の閲覧後にテストに回答したグループAの正答率と、従来手法の閲覧後に回答したグループBの正答率との間で、両側t検定を行った。設問ごとの正答率およびp値を表5.1に示す。小問が設置されているQ8とQ9に関しては、設問全体の正答率と小問ごとの正答率を別途記載した。

グループAとグループB2者の正答率には、ほとんど差は見られなかった。Q6の「骨格画像(ネコ)からの生物を特定する設問」のみ、提案手法に優位傾向($p < 0.1$)が見られた。

表 5.1 テストの結果(*: $p < 0.1$)

設問	正答率(%)		p値
	グループA: 提案手法	グループB: 従来手法	
Q1 ネコ、ヒト、イヌ、ウマの体長順の並び替え	80	70	0.59
Q2 5本指の生物選択	30	10	0.34
Q3 アデロバシレウス、コウモリ、クジラ、ウマの系統樹描画	90	100	0.34
Q4 指骨が最も多い生物の選択	60	60	1.00
Q5 イヌと進化学的に近い生物の選択	60	70	0.73
Q6 骨格画像(ネコ)からの生物特定	30	0	0.08 *
Q7 正しく描画された骨格画像(クジラ)の選択	20	20	1.00
Q8 骨格におけるひじの位置(6問)	95	91	0.68
ウマ	80	90	0.59
クジラ	100	100	1.00
コウモリ	90	90	1.00
アデロバシレウス	100	90	0.34
イヌ	100	90	0.34
ネコ	100	90	0.34
Q9 ヒトと比較した骨格の対応関係(3問)	36	36	1.00
比較対象:コウモリ	70	70	1.00
比較対象:ウマ	0	0	1.00
比較対象:クジラ	40	40	1.00

5.3.2 発見的学習に関する自由記述アンケート

被験者が教材を自由に閲覧している最中に回答させた、自由記述アンケートの結果について以下に記す。

この自由記述アンケートは、教材の閲覧によって、被験者が自ら知識を獲得する「発見的学習」が可能であったかどうかを確かめる目的で行われている。

質問項目は、

1. 教材の閲覧から新たに知ったこと、気づいたこと
2. 教材の閲覧から推察できること
3. 教材の閲覧から疑問に思ったこと

の3点である。回答は、閲覧中に質問に該当する考えがあった場合のみ行われる。

質問項目ごとの回答をおおまかに分類し、以下に記載する。

グループ A(提案手法)とグループ B(従来手法)両者に、手順 4 で同様のアンケートを実施しているが、グループ B の回答結果は「手順 2 の段階で従来手法によって学習した以外で、新たに考えた内容」の記載となる。回答者が属するグループと同様の回答をした人数を合わせて記載する。

1. 教材の閲覧から新たに知ったこと、気づいたこと

< 題材として選択した生物全体に関する内容 >

- ・ 生物の大きさは種によって大きく異なる。(A : 1 名)
- ・ 骨格のサイズ比がわかった。(B : 1 名)
- ・ 生物によってある骨とない骨が存在する。(A : 2 名)
- ・ 骨格の構成は似通っており、他の生物と対応関係がある。(A : 3 名, B : 1 名)
- ・ 7 種の哺乳類では、指の変化意外構造に大きな差がない。(A : 1 名, B : 1 名)
- ・ 生物によって、親指の有無や構造には大きな差がある。(B : 2 名)
- ・ 腕の動かし方によって、肩甲骨の形が変わる。(B : 1 名)
- ・ ヒレや翼を持つ生物の前腕部分は、4 足歩行の生物に比べてねじれが少ない。(B : 1 名)

< 系統樹を参考にしたと考えられる内容 >

- ・ 骨格の形状は、進化の過程で変化している。(A : 1 名)
- ・ 進化の過程で退化する骨がある。(A : 1 名)
- ・ 系統樹上近いとされる生物は、前肢の骨格に大きな差がない。(A : 1 名, B : 1 名)
- ・ 同じ哺乳類でも、進化学的に近い種と遠い種がある。(A : 1 名)
- ・ 哺乳類の祖先から進化する過程で、ヒトには尾が無くなった。(B : 2 名)

- ・どのように哺乳類が進化したのかを知ることができた。(A : 1 名)
- ・クジラ、ウマ、イヌ、ネコはより近い関係にあり意外だった。(A : 1 名)
- ・ネコからウマやイヌが派生している。(A : 1 名)

< 種の比較による内容 >

- ・比較した際、一方にしかない骨がよくわかった。(B : 1 名)
- ・ヒトと哺乳類の祖先とで、腕を構成するパーツは変わらない。(A : 3 名, B : 1 名)
- ・ヒト、ネコ、イヌには、関節部分に小さな骨がある。(B : 2 名)
- ・ネコとイヌを比較すると、骨格の構造にほとんど差が見られない。(B : 3 名)
- ・ネコとイヌは似ているが、微妙に骨格の太さや角度が異なっている。(A : 1 名)
- ・ネコとイヌでは、肩甲骨の角度にすいぶん違いがある。(B : 1 名)
- ・ウマとイヌの外見は似ているが、骨格の構造は大きく異なっている。(B : 2 名)
- ・コウモリとクジラでは、指の数に違いがある。(B : 1 名)
- ・コウモリの羽とクジラのヒレは似ているように感じるが、コウモリの骨格はクジラよりもヒトに近い。(A1)

< 特定の種に着目した内容 >

- ・コウモリの骨格構成は、ほとんどヒトなどと同じである。(A : 3 名, B : 2 名)
- ・ウマには、手にあたる部分の骨が中指 1 本しかない。(B : 4 名)
- ・ウマは、手にあたる部分が他と比較して長い。(B : 1 名)
- ・クジラのヒレ内部にも腕にあたる部分がある。(A : 1 名)
- ・クジラのヒレは、構造的にはヒトなどと同じである。(A : 2 名, B : 2 名)
- ・クジラは他の生物と比べて指骨の数が多い。(A : 1 名, B : 4 名)

2. 教材の閲覧から推察できること

< 教材で提示した生物に関する内容 >

- ・イヌは進化の過程で、関節部分に小さな骨を持つようになったのは、体長が大きくなるために必要だったからではないか。(A : 1 名)
- ・ウマが進化の過程で指の数を減らしたのは、手でなにかをする必要が無くなったからではないか。(A : 1 名)
- ・ウマの足先は、指がくっついたものではないか。(A : 1 名)

< 生物の分類に関する内容 >

- ・似た姿勢で歩いている 4 足歩行の動物は、ネコ〜クジラのグループに属するのではないか。(A : 1 名)
- ・足が退化したクジラは、現在生きている 6 種の中でもより離れたグループに属するのではないか。(B : 1 名)

< 生物の骨格構造に関する内容 >

- ・ どの哺乳類も、外見に大きな差があっても、似通った骨格構造をしているのではないか。(A : 1 名)
- ・ 哺乳類の祖先と現在生きている生物との間には共通点が残っている。(A : 1 名)
- ・ 系統樹で近い関係の種は、骨格の構造も似ているのではないか。(B : 1 名)
- ・ 他の鳥類もコウモリと似た骨格を持っているのではないか。(A : 1 名)
- ・ イヌとネコの骨格が似ているということは、ライオンやトラも同じような骨格なのではないか。(B : 1 名)

< 骨格の発達や退化に関する内容 >

- ・ 手にあたる部分に大きな力がかかる生物は、指の数が少ないのではないか。(B : 1 名)
- ・ 前肢を使用して歩く生物は、指の部分が大きく発達するのではないか。(A : 1 名)
- ・ 生物の動きにしたがって、必要な骨が発達し、不必要な骨が退化するのではないか。(B : 2 名)
- ・ 4 足歩行かつ爪先立ちの生物は、前腕の上腕につながる部分が発達するのではないか。(B : 1 名)

3. 教材の閲覧から疑問に思ったこと

< 骨格構造に関する内容 >

- ・ 進化の過程で、なぜ骨格の変化が起きたのか。(A : 2 名)
- ・ 骨格に大きな差がみられるのは手の先だけで、肩に近い部分にほとんど変化がないのはなぜか。(A : 1 名)
- ・ なぜコウモリの腕は翼という形に姿をかえたのか。(A : 1 名)
- ・ イヌとネコの手首の骨(手根骨)には、なぜ微妙な違いがあるのか。(B : 1 名)
- ・ なぜウマは指の数が減り、ひづめを持つようになったのか。(A : 3 名, B : 1 名)
- ・ ウマの指は、なぜ1本のみが太くなったのか。(A : 1 名, B : 1 名)
- ・ なぜクジラの親指部分が無くなったのか。(A : 2 名)

< その他 >

- ・ なぜクジラはここまで大きく進化したのか。(A : 1 名)
- ・ 肩の骨は、胴体にどのようにつながるのだろうか。(B : 1 名)
- ・ 生物の種によって、筋肉も異なっているのだろうか。(B : 1 名)

5.3.3 印象・操作性に関するアンケート・感想

手順5で行った、提案手法の印象・操作性に関するアンケートによる評価結果と、実験や提案手法に対して寄せられた感想について述べる。

提案手法の印象・評価性に関するアンケートは、5段階評価による質問と、「他に追加した方がよいと思う機能」を問う自由記述項目から成っている。

5段階評価の評価点は、5点：大変満足～1点：大変不満足として付けられている。質問項目ごとの平均評価点と標準偏差を表5-2に記載する。

表 5.2 5段階評価によるアンケート結果

質問項目	平均評価点	標準偏差
1 教材を閲覧してみて、楽しいと感じましたか？	4.75	0.44
2 教材の閲覧により、多角的な視点から知識を得られましたか？	4.25	0.72
3 3DCGモデルの操作(回転, 拡大/縮小, 移動)は簡単でしたか？	4.05	0.83
4 3DCGモデルの操作は直感的に行うことができましたか？	4.35	0.67
5 画面の操作方法を理解することは簡単でしたか？	4.25	0.97
6 骨格の比較は簡単にできましたか？	4.10	0.72
7 生物の大きさの比較は簡単にできましたか？	4.40	0.94
8 骨格の共通点の把握は、簡単にできましたか？	4.00	0.97
9 骨格の差の把握は、簡単にできましたか？	3.95	1.10
10 生物の概形(シルエット)は、2種の3DCGを比較する際の参考になりましたか？	4.20	1.01
11 系統樹は、比較する2種を選択する際の参考になりましたか？	4.45	0.89
12 系統樹は、2種の3DCGを比較する際の参考になりましたか？	4.05	1.05
13 教材の閲覧により、生物への興味・関心が高まりましたか？	4.30	0.57
14 このような3DCGを利用した教材があれば、利用したいと感じますか？	4.80	0.41

自由記述項目である「他に追加した方がよいと思う機能」についての回答は、以下の通りである。複数人から回答のあった項目は、その人数を合わせて記載した。

< 変形アニメーションの再生に関する回答 >

- ・ 変形アニメーションのスピード調節 (4名)
- ・ 変形アニメーションを一コマずつ閲覧する機能

< 3D ビューのカメラ操作に関する回答 >

- ・ 視点のリセット機能(3 名)
- ・ ノーマルモードの注視点をモデルの中心に自動的に設定する機能
- ・ トラッキングモード使用中のズーム機能(2 名)
- ・ 注目した任意の箇所に対するトラッキング機能(5 名)

< 表示や操作性に関する回答 >

- ・ 3次元空間内の特定の軸(例えば縦方向)をわかりやすく表示する機能
- ・ スライダの拡大表示(2 名)
- ・ 3D ビューの操作とスライダの同時操作
- ・ 利き手による UI 表示の変更

< 補足情報に関する回答 >

- ・ 体長に関する詳細な情報(実際の数値など)の記載
- ・ 骨格構造の差に関する解説や説明文(2 名)
- ・ 進化に関する解説や説明文
- ・ 鳴き声を再生する機能
- ・ 各生物の腕の可動範囲などが確認できる機能

実験や提案手法に対して寄せられた感想や要望は、以下の通りである。

< 提案手法の内容に関する内容 >

- ・ 面白かった、楽しかった(4 名)
- ・ 楽器のモデルの違いなどもこれでわかると面白いかなと思う。
- ・ 小さい子などに使ってもらえると興味を持ってもらえるのではないかなと思う。
- ・ クジラやコウモリにも手があると知り、感心した。
- ・ 骨格の変化を追うことで進化の分類がわかりやすくなると思うので、いい教材だと思う。
- ・ 2種類を比べることで、違いを見つけやすいし、アニメーションとして動くことで勉強が嫌いでもとっつきやすく、印象に残りやすいと思った。
- ・ どのように進化したのかが分かり面白かった。系統樹と合わせてみると、近い動物、遠い動物で骨格に差があり、興味深かった。自然とどのような骨格をしているのか、どのように進化したのか、知ることができた。

< 機能に関する感想や要望 >

- ・ トラッキング機能が便利だった。
- ・ 骨格が色付けされていて、大変分りやすかった。
- ・ なかなか狙った場所をみるのが大変だった。
- ・ もっと大きな画面で見たかった(2 名)

第6章 考察

6.1 発見的学習の支援

複合的な視点からの学習

発見的学習に関する自由記述アンケートでは、ほとんどの学習者が「教材(提案手法)の閲覧から新たに知ったこと、気づいたこと」を回答しているほか、提案手法に含まれない内容に関する「推察」や、より発展的な学習につながる様々な「疑問」が寄せられた。

学習者が発見的学習を行った結果であると思えることができる「新たに知ったこと、気づいたこと」の回答には、固有の骨格構造やサイズに関する内容のみでなく、複数の種を比較していくことによって得られた発見が記載されている。それらは、種の類似性・斉一性(骨格構造の共通点や対応関係)や多様性(比較した際の構造の差)の理解につながる知識である。

また、系統樹を参考にしたと考えられる内容では、種の関係性(系統的に見て近い、遠い)を参照しながら、骨格構造の比較を行うことができたことが読み取れる。単なる骨格構造の観察のみでは得難いと考えられる「進化や退化の過程」や、「骨格構造と系統樹の関連性」に関する知識を、学習者が自ら気づきとして得ることができた。系統樹型 UI によるモデル選択を取り入れたことで、骨格構造の変化と進化を結びつけた学習が可能になったと考えられる。

以上より、学習者は提案手法により、生物の進化を垂直的側面(時間次元における適応的变化)と水平的側面(空間次元における多様化)の両者から捉えることが可能であると考えられる。

学習者や閲覧により、提案手法で提示した情報以上の事柄に関する「推察」も行うことができた。その内容は、「教材で示した生物の骨格がなぜその構造をとっているのか」といったものや、「教材で示された以外の生物の骨格構造を予想した」もの、生物の分類や骨格構造の発達・退化に関するものまで多岐に渡る。

教材で示された以外の生物以外の骨格構造を予想したものには、「外見の差に関わらず哺乳類の骨格構造は似通っている」、「系統的に近い関係の種は骨格構造も似ている」、「イヌとネコの骨格が似ているということは、ライオンやトラも同じような骨格である」といったように、実際に正しい知識を推察によって導き出せているものも多い。

また、「進化や退化が生物の生活様式に従って起きている」という内容も、正しい知識を推察できた結果である。提案手法のベースとした教科書に掲載された教材「前肢の骨格比較」が目的としていた、相同器官の多様性・類似性が生物の生活様式に従って生まれたことに関する学習が、提案手法でも可能であることが示された。

教材の閲覧により感じた「疑問」には、骨格構造の変化がなぜ起きたのかに関する内容

のほかに、筋肉や関節の学習につながる内容が寄せられた。

学習者が、教材の内容を超えた疑問を提案手法の閲覧から抱くことが、より発展的な学習への契機になりうると考えられる。

一方で、一部の回答には「ネコからウマやイヌが派生した」といった系統樹を誤って理解してしまった例や、「他の鳥類もコウモリと似た骨格なのではないか」といった分類に対する誤解(コウモリは鳥類ではなく哺乳類である)がみられた。

学習者自らの発見を支援することが本研究のねらいであるため、提案手法では解説的な補足情報を提示していない。しかし割合としては少ないものの、学習において上記のような間違った知識を得てしまう学習者が存在することもわかった。提案手法による自発的な学習ののちに別途解説の機会を設けるなど、正しい知識を伝える考慮が必要であると考えられる。

以上より、提案手法の閲覧によって、学習者は、複合的な視点から生物や進化に関する発見的学習を行うことができると考えられる。その内容は、複数の視点から進化を捉えた「系統樹と骨格構造の関係性」や「進化により相同器官の多様性・類似性が生まれたこと」を含んでおり、進化の教授価値として挙げられた「進化の垂直的側面と水平的側面」の学習につながっている。

3DCG による能動的な観察

自由記述アンケートの回答結果からは、リアルタイム CG を用いた教材の特徴が活かされたと考えられる回答が寄せられ、リアルタイム CG が能動的な観察に適していることが伺えた。特に B グループ(従来手法を先に閲覧したグループ)に属する被験者の回答では、「紙メディアでは得られなかった知識」が記載されているため、よりその傾向が強かった。

関節など特定部分の拡大によって詳細な観察を行ったものや、立体構造を比較した際に骨格構造のねじれや角度に気がついたといった回答が、リアルタイム CG の能動的操作によって獲得された知識である。

拡大や縮小を能動的に行えるリアルタイム CG の特徴が、学習に結びついたと考えられる。また、特定の方向から描かれたイラストからは把握や比較が行いづらい、ねじれや角度といった立体構造の観察に対しても、提案手法が有効であることが示唆された。

6.2 閲覧による知識の定着

評価実験では、被験者を2つのグループに分け、それぞれに提案手法と従来手法による7分間の学習を行わせた後に、教材内容に関するテストを実施している。

テストの結果を分析したところ、設問ごとの正答率には2グループの間でほとんど差がみられなかった。提案手法と従来手法とで知識の定着率に違いは少ないと考えられる。したがって、リアルタイム CG を用いた提案手法は、同程度の内容を持つ紙メディア教材と同様の学習効果が期待できる。

唯一、提案手法に優位傾向がみられた設問が、「骨格画像から生物の種を特定する問

題」である。提案手法と従来手法では、どちらも骨格画像に同様の CG モデルを利用しているが、この設問では正答率に差が生じた。これは、自由記述アンケートの回答からも読み取れるように、提案手法が立体構造の観察に適していることに起因していると予想される。

6.3 学習者の感情

学習に対する楽しさ

アンケートによる印象への評価では、「閲覧に対する楽しさ(平均評価点：4.75)」や「提案手法のような教材があったら利用したい(平均評価点：4.80)」という質問に対して、特に高い点数が付けられた。感想でも、面白かった、楽しかったという意見が多数寄せられた。

楽しさにつながった要素としては、感想に記載された「比較することで違いが見つけやすく、印象に残る」「アニメーションとして動くことでとっつきやすい」といった内容が挙げられる。リアルタイム CG を使った変形アニメーションを用いた比較方法が、高評価の要因であると考えられる。

また、「系統樹と合わせてみることで骨格との関係性がわかり興味深い」「閲覧によってわかった内容に驚いた」といった感想から、「学習者が自発的に発見を行うこと」が学習の楽しさにつながったと予想される。

学習による生物への興味・感心の向上

「教材の閲覧により、生物への興味・関心が高まりましたか？」という質問項目の平均評価点は 4.30 点となった。これは、筆者らが先行研究[29]で行った同様の質問に対する平均評価点 4.10 点より高く、提案手法がより生物への興味を喚起できる可能性があることを示している。

先行研究では骨格構造のみを題材として扱っており、限定的な情報の提示が低い評価につながったと考えられた。

本研究の提案手法では、系統樹型 UI の表示により、学習者が閲覧によって得られる知識や考慮できる内容が大きく増えている。系統樹の表示により、新たな側面の発見を促すことができた点が、高評価へつながったと考えられる。

上記のような学習者の感情は、提案手法が題材とする内容を超えた、発展的な学習へとつながる契機となりうるものである。

6.4 操作性

タブレット端末の使用について

評価実験はタブレット端末の中でも比較的画面サイズが小さい iPad mini を用いて行ったが、より大きな画面で閲覧を行いたいという意見が寄せられた。iPad mini は教育の現場で使われることある機種であるため、画面サイズに関わらず快適な閲覧を実現するためには、ズーム機能や UI 表示サイズの調節機能等で画面サイズの不足を補う必要があると考えられる。

また、タブレット端末はタッチによって操作が行われるため、利き手に考慮した UI のカスタマイズへの要望が寄せられた。提案手法では、変形アニメーションの調節スライダを操作しながら 3D ビューを見ようとした際に、手で画面が隠れてしまうケースが見られた。左手で操作を行えば、画面を遮らずに操作を行うことが可能であるが、ほとんどの被験者は利き手のみを使って教材を閲覧する傾向がみられることが分かった。

タブレット端末を用いた教材では、上記に考慮した機能を実装することで、より快適な閲覧を実現できると考えられる。

画面およびカメラの操作

アンケート結果では、「3DCG モデルの直感的な操作」や「画面の操作方法の理解」に対する平均評価点が 4.25 以上と高い値となっている。3DCG モデルの操作は直感的に行うことが可能であり、画面全体の操作方法の把握は容易であったと考えられる。

一方で、「3DCG モデルの操作(回転、拡大/縮小、移動)は簡単でしたか?」という項目は若干低い値をとっており、感想や「追加すべき機能」でもカメラ操作に関する困難さが伺える。

したがって、3点までを使うタッチ操作が直感的に行うことができたとしても、学習者が 3D ビューを自在に閲覧できるようになるまでには、ある程度の慣れが必要であると考えられる。

用途によって使い分けることができるカメラモードの用意については、便利だったという意見が得られたものの、トラッキングモードでカメラの動作を制限したことへの不満が伺える。

3DCG の操作を簡単にするためにカメラの移動を制限(特定の方向にしか回転しない、カメラの高さが決まっているなど)する方法は、3D アプリケーションでしばしば用いられる手法であるが、観察を目的とするコンテンツでは高い自由度を持つ操作性が求められる。高い操作性と、閲覧における自由度、両者のバランスを探りながら改良を重ねる必要があると考えられる。

6.5 機能

変形アニメーションの提示

提案手法では、サイズの異なる様々なモデルを変形アニメーションによって比較している。変形アニメーションによる比較が、自発的な発見を促したことは先に述べたが、現在のアニメーション提示手法に付随した機能を求める声もあった。具体的には、アニメーションのスピードやコマを制御する機能や、特定の位置に注視しながらアニメーションを再生する機能である。

比較対象である2種間の変形アニメーションは、CGモデルの頂点を線形補完によって移動しながらレンダリングを繰り返すことで再生されている。このため、サイズが大きく異なるモデル(例：アデロバシレウスとクジラ)を比較対象として選択してアニメーションを再生した場合、再生におけるほとんどの段階で、モデルの形状はサイズの大きい種(クジラ)の状態に変化してしまう。そのような比較を行った場合、アニメーションの再生中に、形状の変化が大きく起こるタイミングが限られてしまう。サイズ差の著しいモデル間において、モーフィング処理による線形補完アニメーションには問題点があることが示唆された。アニメーションのスピードやコマを制御する機能の要望は、この問題点から寄せられた回答である。

この問題点は、要望のあった機能の実装のほかに、モデル比にしたがってツール側でアニメーション速度を自動的に調節して再生に反映することで回避できる可能性がある。

また、トラッキング機能は閲覧の補助機能として有効であったとみられるが、任意の場所に注視しながら変形アニメーションを閲覧したいという要望も寄せられた。変形アニメーションの注視点を任意に設定することができれば、より簡単に特定箇所の観察を行うことができると考えられる。この機能を追加することで、高い操作性と自由度の両者を持つコンテンツの開発が可能になると予想される。

補足情報の提示について

追加機能についての質問では、「骨格や系統樹に関する解説や説明」、「体長、鳴き声、生物の動き等情報の追加」を求める意見もみられた。

学習者の自発的な発見を促す目的で教材を開発しているため、提示される情報に制限をかけることは必要であると考えられるが、教材の閲覧により推察できた内容や疑問に感じた内容をより発展的な学習へつなげていくためには、補足情報を提示する必要がある。

誤った知識の獲得を防ぐためにも、提案手法による自発的な学習後に解説の機会を設けるなど、生物や進化の学習全体を通した提案手法の使用方法について、検討を重ねる必要があると考えられる。

第7章 まとめ

7.1 結論

本研究では、進化への自発的な発見を促す新たな教材として、3DCG モデルの変形アニメーションと系統樹型の UI を用いた、リアルタイム CG 骨格比較ツールを提案した。

リアルタイム CG やタブレット端末による ICT 教育が普及したことで実現が可能となったデジタルツールは、学習者が能動的に操作を行うことが可能であるため、学習者が自発的に知識を獲得する発見的学習に適している。また、WebGL を利用することで、OS に縛られない汎用性の高い教材を開発できる。

開発したツールは、その現象が長期的な変化の結果生ずるものであること、実験的に検証するのが不可能であることから教材の不足が指摘されている、生物の「進化」に関する教材である。進化を理解するために必要な「系統樹」「骨格の観察」を主軸とし、学習者が自らの観察によって発見的学習が可能となる手法を開発した。

提案手法では、進化教材として取り上げられることの多い「前肢の骨格比較」を題材として選定しており、系統樹を参考とした CG モデルの骨格比較により、進化への理解を促すことを目的としている。学習者は比較ツールを使用する際に、系統樹を模した系統樹型 UI から比較対象となる 2 種の生物を選択し、CG モデルの変形アニメーションの閲覧から 2 種の比較を行う。このツールにより、学習者は「系統樹の学習」、「骨格構造の観察」、「骨格の比較による自発的な知識の獲得」という 3 つの観点から進化について学習を行うことが可能になる。複数の観点から進化を学ぶことで、学習者は複合的な視点から、能動的に進化を捉えることができると予想した。

評価実験の結果から、学習者は提案手法の閲覧により、複合的な視点から生物や進化に関する発見的学習を行うことができたと考えられる。発見の内容は、複数の視点から進化を捉えた「系統樹と骨格構造の関係性」や「進化により相同器官の多様性・類似性が生まれたこと」を含んでおり、進化の教授価値として挙げられた「進化の垂直的側面と水平的側面」の学習につながる内容である。今後の学習への契機となる、学習による楽しさや、生物への興味・関心の向上についても高い評価を得ることができた。

また、提案手法は従来の紙メディアと同程度の知識を学習者に与えることが可能であるほか、立体構造の観察に優位であることが示唆された。特に、提案手法は、骨格構造のねじれや角度の比較といった、立体構造の比較に対して有効であると考えられる。

今後、提案手法の操作性や自由度を調整していくことで、より快適な観察を実現できると考えられる。また、提案手法の使用による発見的学習の後に解説の機会を設けるなど、生物や進化の学習全体を通した学習方法について、検討を重ねる必要があると考えられる。

7.2 展望

本研究の評価実験により、提案手法が立体構造の観察や比較において、特に有効であることが示唆された。変形アニメーションによる立体構造の比較は、教材に限らず多くの場面で用いられる可能性のある情報の提示手法であると考えられる。

謝辞

本論文を執筆するにあたり、金尚泰先生と研究室の皆さまには、幅広い観点から熱心にご指導をいただきました。大学生活からの長きに渡り、研究へのアドバイスのみならず、様々な場面でご助力くださったことに、改めて深謝いたします。

お忙しい中快く実験にご協力いただいた方々や、森田ひろみ先生をはじめ研究に対して新鮮な意見をくださったみなさまにも、大変お世話になりました。心より感謝いたします。

日頃多くの面で私を支えてくれた家族と友人にも、この場を借りて深く感謝申し上げます。

最後に、大学院生活の大半をそばで支えてくれたハムスターのメメタァと、なにより私に最高の学びを与えてくれた良き理解者である父に、感謝の言葉を記して謝辞といたします。

参考文献

- [1]アンドリアス・アンユル, 吉川邦夫訳. 実践プログラミングWebGL, 2012, 翔泳社, pp.318
- [2]飯島貴志. CG デザイナーのためのグラフィックバイブル 人体のしくみ. ワークスコーポレーション, 2003, pp.200
- [3]NHK「恐竜」プロジェクト編. "恐竜vsほ乳類 : 1億5千万年の戦い : ビジュアル版". ダイヤモンド社, 2006, pp.94
- [4]Ellenberger, w., et al. An atlas of animal anatomy for artists, 1956, Dover Publications, pp.151
- [5]太田聡. “科学的な思考力・判断力・表現力を高める理科学習の展開 : 医学と連携した科学教育の礎を築く理科教材開発の研究”. 滋賀大学教育学部附属中学校研究紀要. 2015, 第57集, pp.44-51
- [6]大槻説乎. “発見的学習とその支援環境”. 1993, 人工知能学会誌 8(4), pp.411-418
- [7]“外務省: 生物多様性条約(生物の多様性に関する条約:Convention on Biological Diversity (CBD))”. 外務省. 2015, <http://www.mofa.go.jp/mofaj/gaiko/kankyo/jyoyaku/bio.html> (2016/12/29)
- [8]”教育の情報化ビジョン”. 文部科学省. 2011, http://www.mext.go.jp/b_menu/houdou/23/04/_icsFiles/afieldfile/2011/04/28/1305484_01_1.pdf (2016/12/29)
- [9]神谷敏郎. “骨の動物誌”. 東京大学出版社, 1995, pp.177
- [10]”Creature Animation”. Emily Willoughby. http://creature.kestrelmoon.com/demo_raptor.html (2017/1/8)
- [11]”高等学校学習指導要領解説 理科編”. 文部科学省. 2009, http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afieldfile/2010/01/29/1282000_6.pdf (2016/12/29)
- [12]金井康恭, 小池 啓一. “高校生物IIにおける「生物の進化」の取扱い”. 群馬大学教育実践研究. 2012, (29), pp.45-49
- [13]佐藤崇之, 大鹿聖公, “教科書分析と教材研究から見た高等学校生物における進化の単位に関する一考察”. 広島大学大学院教育学研究科紀要. 2005, 第二部 文化教育開発関連領域 (54), pp.17-24
- [14] Sandvik, H. “Tree thinking cannot taken for granted: challenges for teaching phylogenetics”. Theory Biosci. 2008, 127(1), pp.45–51.
- [15]塩野貴之他. “オンラインツール"TimeTree: the timescale of life"を用いた生物多様性科学の授業開発とその評価”. 理科教育学研究. 2014, 54(3), pp.427-437
- [16]”three.js webgl - geometry - large mesh”. Ricardo Cabello. http://pages.cs.wisc.edu/~lizzy/mrdoob-three.js-ef5f05d/examples/webgl_geometry_large_mesh.html (2017/1/8)

- [17]瀬戸崎典夫他. "彫像観賞用AR教材における能動的操作の有用性に関する検討". 日本教育工学会論文誌, 2011, 35(Suppl.), pp.105-10
- [18]塚田捷他. "未来へひろがるサイエンス 2". 新興出版社啓林館, 2016, pp.270
- [19]中貴俊他. "スマートデバイスによる3DCGを用いた電子教材の提示・操作に関する研究 : タブレット端末3DCGコンテンツ開発に向けたフレームワーク構築". 情報文化学会誌. 2013, 20(2), pp.19-426
- [20]野口孝文, 田中譲. "実世界と相互作用するマイクロワールド", 1995, 人工知能学会誌 10(3), pp.383-392
- [21]Hedges, S.B., Dudley, J., & Kumar, S. "TimeTree: A public knowledge-base of divergence times among organisms". Bioinformatics. 2006, 22, pp.2971-2972
- [22]福井智紀, 鶴岡義彦. "高校生物II教科書における生物の進化の取扱い : 進化的内容の90カテゴリーとISM法を用いた分析". 千葉大学教育学部研究紀要. 2000, I, 教育科学編 48, pp.75-93
- [23]福井智紀, 鶴岡義彦. "生物進化の教授価値の検討 : 我が国と米国における諸議論を手掛かりに". 千葉大学教育学部研究紀要. 2002, 50, pp.69-82
- [24]"平成27年度 学校における教育の情報化の実態等に関する調査結果 : 文部科学省". 文部科学省.
http://www.mext.go.jp/component/a_menu/education/micro_detail/__icsFiles/afieldfile/2016/10/13/1376818_1.pdf (2016/12/28)
- [25]松田晃一. "WebGL+HTML5 3DCG プログラミング入門". カットシステム, 2012, pp.574
- [26]松原行宏他. "バーチャルリアリティ環境下での教科教育を考慮した発見的学習支援機構". 電子情報通信学会論文誌. 2000, D-I, 情報・システム, I-情報処理 J83-D-1(10), pp.1109-1119
- [27]"学びのイノベーション事業実証研究報告書". 文部科学省. http://jouhouka.mext.go.jp/school/pdf/manabi_no_innovation_report.pdf (2016/12/28)
- [28]Rowe, T. S. "Definition, diagnosis, and origin of Mammalia". Journal of Vertebrate Paleontology. 1988, 8(3), pp.241-264
- [29]岑村春香, 金尚泰. "形態比較のためのリアルタイムCG骨格標本の提案". 日本デザイン学会. 第62回研究発表大会概要集, 2015, D9-02
- [30]山森一人他. "HTML5 を用いた天体シミュレーションによる小学生理科補助教材の開発". 宮崎大 学工学部紀要. 2014, 43, pp.189-193
- [31]吉富健一. "WebGLを用いた天体学習用3D教材の開発". 広島大学大学院教育学研究科紀要. 2015, 第二部, 文化教育開発関連領域 (64), pp.9-16

付録

付録として、以下を添付する。

- 1, ソースコード：リアルタイム骨格比較ツール
 - ・ index.html (比較ツールの Web ページ本体および WebGL の実装)
 - ・ style.css (Web ページのスタイルシート)
 - ・ gt.js (系統樹型 UI の実装)
 - ・ skeleton.js (骨格モデルデータ。データ形式の判別に必要な部分のみを掲載する)
- 2, 評価実験に使用した書類
 - ・ 評価実験の説明に用いた資料：1 ページ
 - ・ 従来手法(紙メディア)の教材：3 ページ
 - ・ 教材に関するテスト：4 ページ

```

<!-- This code use Three.js r66 -->
<!DOCTYPE html>
<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
  <title>WebGL Skeleton Viewer</title>
  <script type="text/javascript" src="js/libs/jquery-1.12.3.js"></script>
  <script type="text/javascript" src="js/libs/jquery-ui.min-v1.12.1.js"></script>
  <script type="text/javascript" src="js/libs/three.min.js"></script>
  <script type="text/javascript" src="js/controls/OrbitControls.js"></script>
  <link rel="stylesheet" type="text/css" href="css/style.css" media="all" />
  <script src="js/gt.js" type="text/javascript" charset="utf-8"></script>
  <script src="js/libs/helvetiker_bold.typeface.js"></script>
  <script type="text/javascript">
    window.mod_pagespeed_start = Number(new Date());
  </script>
</head>

<body>
  <div id="ctrl">
    <h1>Skeleton Viewer</h1>
    <h2>哺乳類の骨格比較 -前肢-</h2>
    <div id="sliderControl">
      <a href="#" id="reset-blend-targets0" class="button1"></a>
      <input type="range" id="morphRange" min="0" max="1" value="0" step="0.01" />
      <a href="#" id="reset-blend-targets1" class="button2"></a>
      <input type="hidden" id="rangeValue">
      <input type="hidden" id="MorphTarget1">
      <input type="hidden" id="MorphTarget2">
      
      
      
      
      <div style="display: block; float: left; border: thin solid #666666; border-radius: 1px; padding: 3px 3px;">
        
        
      </div>
    </div>
    
  </div>

  <div id="gt">
    <div id="GT_Main">
      
      <div id="GT_1">
        
      </div>
      <div id="GT_2">
        
      </div>
      <div id="GT_3">
        
        
        
        
        
        
      </div>
      <div id="GT_4">
        
      </div>
      <div id="GT_5">
        
      </div>
    </div>
  </div>

  <script type="text/javascript">

```

```

// Ready function
$(document).ready(
  function(){
    // Reset Slider Button
    $("#reset-blend-targets0").click(
      function(){
        morphReset(0);
        if(AutoMorphing == true)
          setAutoMorphing();
      });

    $("#reset-blend-targets1").click(
      function(){
        morphReset(1);
        if(AutoMorphing == true)
          setAutoMorphing();
      });

    $("#btn1").click(
      function(){
        setAutoMorphing();
      });

    $("#btn2").click(
      function(){
        setAutoRotation();
      });

    $("#btn3-1").click(
      function(){
        changeTexture();
      });

    $("#btn3-2").click(
      function(){
        skComp();
      });

    $("#btn3-3").click(
      function(){
        skComp();
      });

    // Relate Slider and Morphing Function
    $("#morphRange").on("input change", function(){
      slideRange(this.value);
    });
  }
);

// Set WebGL Variables
var camera, camera2, cameraD, scene, scene2, scene3, sceneD, projector,renderer;
var mesh1_bone =new THREE.Object3D();
var loader=new THREE.JSONLoader();
var loader2=new THREE.JSONLoader();
var texture = new THREE.ImageUtils.loadTexture('model/Skeleton_path6/color.png');
var material = new THREE.MeshLambertMaterial({map:texture});
var material2 = new THREE.MeshBasicMaterial({color: 0x444444});
var material3 = new THREE.LineBasicMaterial({ color: 0xaaaaaa});

var material4 = new THREE.LineBasicMaterial({ color: 0x333333});
var material4_w2 = new THREE.LineBasicMaterial({ color: 0x333333, linewidth: 2});
var material4_w3 = new THREE.LineBasicMaterial({ color: 0x333333, linewidth: 3});

var mouse={x:0,y:0};
var SHADOW_MAP_WIDTH=1024,SHADOW_MAP_HEIGHT=1024;
var resettoCenterCheck=false;
var toggleCoords={clientX:0,clientY:0};
var renderReady=0;

```

```

// var SCREEN_WIDTH=window.innerWidth;
var SCREEN_WIDTH = 1368;
var SCREEN_HEIGHT=window.innerHeight;
var FLOOR=0;
var NEAR=5,FAR=2000;
var sceneHUD,cameraOrtho,hudMaterial;
var initPos=new THREE.Vector3(700,50,1900);
var initLight=new THREE.Vector3(0,1500,1000);
var initRot = -90*(Math.PI / 360);
var deltaCam=new THREE.Vector3();
var ablight,ablight2,dlight,d2light,d3light;

var counter = 0;
var AutoMorphing = false;
var AutoRotation = false;
var Colorize = false;
var SkeletonComparing = false;
var moveCounter = 0;
var panCounter = 0;

var Animation = true;
var AnimationSpeed = 0.025;
var StopTime = 25;
var StopAnimation = 20;

var mj = 150; //measure(Scale) unit
var mJx2 = 300; //measure(Scale) unit
var mJX = 20;
var mJY = 15;
var mJZ = 40;

var rV = 0.5; //rangeValue
var mt1, mt2; //morphTarget
var tCamX, tCamY, tCamZ; //tmp Camera position
var tRot; //tmp Rotation
var tmpAM; //tmp AutoMorphin

// taget for Tracking camera mode
var targetArray = [ -262, 80, -150, -180, -150, -60, 210];
var targetArray3D = [ [50, -262, 50], [500, 80, 500], [280, -150, 280], [280, -180, 280], [340, -150, 340], [580, -60, 580], [2000, 210, 2000]];
var targetDelta = 0;
var targetDef;
var targetDelta3D = [0, 0, 0];
var targetDef3D = [0, 0, 0];
var targetErr3D = [false, false, false]
var targetErr = false;
var targetErr2 = false;

var ErrCounter = 0;
var SliderMoveR = true;

// Create objects for Scale
var lineGeometryX;
var lineX = new THREE.Object3D();

var lineGeometryY;
var lineY = new THREE.Object3D();

var lineGeometryZ;
var lineZ = new THREE.Object3D();

for(var i=0; i<=mJX/2; i++) {
    lineGeometryX = new THREE.Geometry();
    if(-1500+i*mJx2 == 0){
        lineGeometryX.vertices.push(
            new THREE.Vector3( -1500, -275, 0 ),
            new THREE.Vector3( 1500, -275, 0 )
        );
        var line_x = new THREE.Line( lineGeometryX, material3 );
    }
}

```



```

    } else {
        lineGeometryX.vertices.push(
            new THREE.Vector3( ((-0.5)*mjX/2+i)*mJx2, -275, -10 ),
            new THREE.Vector3( ((-0.5)*mjX/2+i)*mJx2, -275, 10 )
        );
        var line_x = new THREE.Line( lineGeometryX, material3 );
    }
    lineX.add(line_x);
}
lineX.rotation.y=initRot;

for(var i=0; i<=mjY/2; i++) {
    lineGeometryY = new THREE.Geometry();
    if( i == 0){
        lineGeometryY.vertices.push(
            new THREE.Vector3( 0, -275, 0 ),
            new THREE.Vector3( 0, 1500, 0 )
        );
        var line_y = new THREE.Line( lineGeometryY, material3 );
    } else {
        lineGeometryY.vertices.push(
            new THREE.Vector3( 0, -275+i*mJx2, -10 ),
            new THREE.Vector3( 0, -275+i*mJx2, 10 )
        );
        var line_y = new THREE.Line( lineGeometryY, material3 );
    }
    lineY.add(line_y);
}
lineY.rotation.y=initRot;

for(var i=0; i<=mjZ/2; i++) {
    lineGeometryZ = new THREE.Geometry();
    if(-3000+i*mJx2 == 0){
        lineGeometryZ.vertices.push(
            new THREE.Vector3( 0, -275, -3000 ),
            new THREE.Vector3( 0, -275, 3000 )
        );
        var line_z = new THREE.Line( lineGeometryZ, material3 );
    } else {
        lineGeometryZ.vertices.push(
            new THREE.Vector3( 0, -285, ((-0.5)*mjZ/2+i)*mJx2),
            new THREE.Vector3( 0, -265, ((-0.5)*mjZ/2+i)*mJx2)
        );
        var line_z = new THREE.Line( lineGeometryZ, material3 );
    }
    lineZ.add(line_z);
}

lineZ.rotation.y=initRot;

var Texts3 = new THREE.Object3D();

var TextGeometry = new THREE.TextGeometry( ' 1 m', {
    size: 30, height: 1, curveSegments: 3,
    font: "helvetiker", weight: "bold", style: "normal"
});
var Text1m = new THREE.Mesh( TextGeometry, material3 );
Text1m.position.y = 10;
Texts3.add(Text1m);

var TextGeometry = new THREE.TextGeometry( ' 2 m', {
    size: 30, height: 1, curveSegments: 3,
    font: "helvetiker", weight: "bold", style: "normal"
});
var Text2m = new THREE.Mesh( TextGeometry, material3 );
Text2m.position.y = 310;
Texts3.add(Text2m);

var TextGeometry = new THREE.TextGeometry( ' 3 m', {

```

```

        size: 30, height: 1, curveSegments: 3,
        font: "helvetiker", weight: "bold", style: "normal"
    });
    var Text3m = new THREE.Mesh( TextGeometry, material3 );
    Text3m.position.y = 610;
    Texts3.add(Text3m);

    var TextGeometry = new THREE.TextGeometry( ' 4 m', {
        size: 30, height: 1, curveSegments: 3,
        font: "helvetiker", weight: "bold", style: "normal"
    });
    var Text4m = new THREE.Mesh( TextGeometry, material3 );
    Text4m.position.y = 910;
    Texts3.add(Text4m);

    var TextGeometry = new THREE.TextGeometry( ' 5 m', {
        size: 30, height: 1, curveSegments: 3,
        font: "helvetiker", weight: "bold", style: "normal"
    });
    var Text5m = new THREE.Mesh( TextGeometry, material3 );
    Text5m.position.y = 1210;
    Texts3.add(Text5m);

    Texts3.rotation.y=-initRot

// Run Initializing Function
init();

setInterval(loop,1000/24);

// Initializing Function
function init(){

    container=document.createElement("div");
    document.body.appendChild(container);
    loadingmodal=document.getElementById("whatsloaded");

    camera=new THREE.PerspectiveCamera(40,1368/window.innerHeight,0.1,10000);
    camera.position.set(120, 115, 985);
    camera.rotation.set(-0.1, 0.25, 0.05);

    camera2=new THREE.PerspectiveCamera(40,1368/window.innerHeight,0.1,10000);
    camera2.position.set(120, 115, 985);
    camera2.rotation.set(-0.1, 0.25, 0.05);

    cameraD=new THREE.PerspectiveCamera(40,1368/window.innerHeight,1,2);

    // Create Scene
    scene=new THREE.Scene();
    scene2=new THREE.Scene();
    scene3=new THREE.Scene();
    sceneD=new THREE.Scene();

    // Light setting
    ablight = new THREE.AmbientLight(0x444433);
    scene.add(ablight);

    dlight=new THREE.DirectionalLight(0xfffff,0.8);
    dlight.position.x=90;
    dlight.position.y=100;
    dlight.position.z=30;
    scene.add(dlight);

    d2light=new THREE.DirectionalLight(0xfffff,0.3);
    d2light.position.x=-90;
    d2light.position.y=-100;
    d2light.position.z=-30;

```

```

scene.add(d2light);

d3light=new THREE.DirectionalLight(0xfffff,0.3);
d3light.position.x=-90;
d3light.position.y=-120;
d3light.position.z=-120;
scene.add(d3light);

var d4light=new THREE.DirectionalLight(0xfffff,0.4);
d4light.position.x=20;
d4light.position.y=150;
d4light.position.z=100;
scene.add(d4light);

// Create Renderer
renderer=new THREE.WebGLRenderer({clearColor:0,clearAlpha:1,antialias:true});
renderer.setSize(SCREEN_WIDTH,SCREEN_HEIGHT);
renderer.domElement.style.position="relative";
container.appendChild(renderer.domElement);
renderer.setClearColor(0x000000,1);
renderer.autoClear=false;

// Add Renderer to DOM
container.appendChild(renderer.domElement);

// Load models
loader.load('model/Skeleton_path6/skeleton.js', loadCallBack);
loader2.load('model/Skin2_path6/skin_re.js', loadCallBack2)

setRange(0.5);

scene3.add(lineX);
scene3.add(lineY);
scene3.add(lineZ);
scene3.add(Texts3);
}

// Callback function for Skeleton model
function loadCallBack(geometry){

    mesh1_bone=new THREE.Mesh(geometry, material);
    geometry.computeVertexNormals();
    mesh1_bone.material.shading=THREE.SmoothShading;
    mesh1_bone.material.morphTargets=true;

    mesh1_bone.scale.set(55,55,55);
    mesh1_bone.position.x=0;
    mesh1_bone.position.y=-275;
    mesh1_bone.position.z=0;
    mesh1_bone.rotation.y=initRot;
    mesh1_bone.doubleSided=true;
    mesh1_bone.castShadow=true;
    mesh1_bone.receiveShadow=true;
    mesh1_bone.frustumCulled = false;

    scene.add(mesh1_bone);
}

// Callback function for Skin model
function loadCallBack2(geometry){

    mesh2_skin=new THREE.Mesh(geometry, material2);

    mesh2_skin.material.morphTargets=true;

    mesh2_skin.scale.set(55,55,55);
    mesh2_skin.position.x=0;
    mesh2_skin.position.y=-275;
    mesh2_skin.position.z=0;

```

```

    mesh2_skin.rotation.y=initRot;
    mesh2_skin.doubleSided=true;
    mesh2_skin.castShadow=true;
    mesh2_skin.receiveShadow=true;
    mesh2_skin.frustumCulled = false;

    scene2.add(mesh2_skin);
}

// if user slide the range
function slideRange(value){

    AutoMorphing = true;
    setAutoMorphing();

    if(mt1 == 0) {
        mesh1_bone.morphTargetInfluences[mt2-1] = value;
        mesh2_skin.morphTargetInfluences[mt2-1] = value;
    } else {
        mesh1_bone.morphTargetInfluences[mt1-1] = 1-value;
        mesh2_skin.morphTargetInfluences[mt1-1] = 1-value;
        mesh1_bone.morphTargetInfluences[mt2-1] = value;
        mesh2_skin.morphTargetInfluences[mt2-1] = value;
    }

    if(targetErr3D[0] == false) {
        camera2.position.x = targetDef3D[0] + (targetDelta3D[0]*value);
        camera2.position.z = targetDef3D[2] + (targetDelta3D[2]*value);
    } else {
        camera2.position.x = targetDef3D[0] - (targetDelta3D[0]*value);
        camera2.position.z = targetDef3D[2] - (targetDelta3D[2]*value);
    }

    if(targetErr3D[1] == false) {
        controls2.target.y = targetDef + (targetDelta*value);
        camera2.position.y = targetDef3D[1] + (targetDelta3D[1]*value);
    } else {
        controls2.target.y = targetDef - (targetDelta*value);
        camera2.position.y = targetDef3D[1] - (targetDelta3D[1]*value);
    }
    rV = value;
    setRange(value);
}

// reset morphTargetInfluences
function morphReset(value) {

    setAutoMorphing();

    Animation = true;
    StopAnimation = 20;

    ErrCounter = 0;
    SliderMoveR = true;

    if(value == 0) {
        if(mt1 == 0) {
            mesh1_bone.morphTargetInfluences[mt2-1] = 0;
            mesh2_skin.morphTargetInfluences[mt2-1] = 0;
        } else {
            mesh1_bone.morphTargetInfluences[mt1-1] = 1;
            mesh2_skin.morphTargetInfluences[mt1-1] = 1;
            mesh1_bone.morphTargetInfluences[mt2-1] = 0;
            mesh2_skin.morphTargetInfluences[mt2-1] = 0;
        }
        rV = 0;
    } else if(value == 1) {
        if(mt1 == 0) {
            mesh1_bone.morphTargetInfluences[mt2-1] = 1;

```

```

        mesh2_skin.morphTargetInfluences[mt2-1] = 1;
    } else {
        mesh1_bone.morphTargetInfluences[mt1-1] = 0;
        mesh2_skin.morphTargetInfluences[mt1-1] = 0;
        mesh1_bone.morphTargetInfluences[mt2-1] = 1;
        mesh2_skin.morphTargetInfluences[mt2-1] = 1;
    }
    rV = 1;
}

if(targetErr3D[0] == false) {
    camera2.position.x = targetDef3D[0] + (targetDelta3D[0]*value);
    camera2.position.z = targetDef3D[2] + (targetDelta3D[2]*value);
} else {
    camera2.position.x = targetDef3D[0] - (targetDelta3D[0]*value);
    camera2.position.z = targetDef3D[2] - (targetDelta3D[2]*value);
}
if(targetErr3D[1] == false) {
    controls2.target.y = targetDef + (targetDelta*value);
    camera2.position.y = targetDef3D[1] + (targetDelta3D[1]*value);
} else {
    controls2.target.y = targetDef - (targetDelta*value);
    camera2.position.y = targetDef3D[1] - (targetDelta3D[1]*value);
}
}

function setAutoMorphing() {
    if(counter!=0){
        if(AutoMorphing == true) {
            AutoMorphing = false;
            $("#btn1").css({opacity: "0.4"});
        } else if(AutoMorphing == false) {
            AutoMorphing = true;
            // StopAnimation = 20;
            $("#btn1").css({opacity: "1.0"});
        }
    }
}

function setAutoRotation() {
    if(counter!=0){
        moveCounter = 0;
        if(AutoRotation == true) {
            AutoRotation = false;
            $("#btn2").css({opacity: "0.4"});
        } else if(AutoRotation == false) {
            AutoRotation = true;
            $("#btn2").css({opacity: "1.0"});
        }
    }
}

function changeTexture() {
    if(counter!=0){
        if(Colorize == true) {
            Colorize = false;
            $("#btn3-1").attr("src","img/ctrl_button3-12.png");
            mesh1_bone.material.map = new THREE.ImageUtils.loadTexture("model/Skeleton_path6/color.png");
        } else if(Colorize == false) {
            Colorize = true;
            $("#btn3-1").attr("src","img/ctrl_button3-1.png");
            mesh1_bone.material.map = new THREE.ImageUtils.loadTexture("model/Skeleton_path6/0" + mt1 + "-0" + mt2 + ".png");
        }
    }
}

function skComp() {
    if(counter!=0){

```

```

if(SkeletonComparing == true) {
    mousedown = false;
    SkeletonComparing = false;
    $("#btn3-3").css({opacity: "0.4"});
    $("#btn3-2").css({opacity: "1.0"});
    $("#plus").css({opacity: "1.0"});

    camera.position.set(tCamX, tCamY, tCamZ);

    mesh1_bone.rotation.y = tRot;
    mesh2_skin.rotation.y = tRot;
    lineX.rotation.y = tRot;
    lineY.rotation.y = tRot;
    lineZ.rotation.y = tRot;
    Texts3.rotation.y = -tRot;

} else if(SkeletonComparing == false) {
    SkeletonComparing = true;

    $("#btn3-2").css({opacity: "0.4"});
    $("#btn3-3").css({opacity: "1.0"});
    $("#plus").css({opacity: "0.0"});

    tCamX = camera.position.x;
    tCamY = camera.position.y;
    tCamZ = camera.position.z;

    tRot = mesh1_bone.rotation.y;
}
}
}

function loop(){

    if (AutoMorphing == true && StopAnimation == StopTime) {
        if (Animation == true && mesh1_bone.morphTargetInfluences[mt2-1] <= 1 && SliderMoveR ==true) {

            if(mt1==0){
                mesh1_bone.morphTargetInfluences[mt2-1] += AnimationSpeed;
                mesh2_skin.morphTargetInfluences[mt2-1] += AnimationSpeed;
            } else {
                mesh1_bone.morphTargetInfluences[mt1-1] -= AnimationSpeed;
                mesh1_bone.morphTargetInfluences[mt2-1] += AnimationSpeed;
                mesh2_skin.morphTargetInfluences[mt1-1] -= AnimationSpeed;
                mesh2_skin.morphTargetInfluences[mt2-1] += AnimationSpeed;
            }

            if(canMorph) {
                if(targetErr3D[0] == false) {
                    camera2.position.x += targetDelta3D[0]/40;
                    camera2.position.z += targetDelta3D[2]/40;
                } else {
                    camera2.position.x -= targetDelta3D[0]/40;
                    camera2.position.z -= targetDelta3D[2]/40;
                }
                if(targetErr3D[1] == false) {
                    controls2.target.y += targetDelta/40;
                    camera2.position.y += targetDelta3D[1]/40;
                } else {
                    controls2.target.y -= targetDelta/40;
                    camera2.position.y -= targetDelta3D[1]/40;
                }
            }

            ErrCounter = 0;

        } else if (1 < mesh1_bone.morphTargetInfluences[mt2-1]){
            Animation = false;
            SliderMoveR = false;

```

```

if(mt1==0){
    mesh1_bone.morphTargetInfluences[mt2-1] = 1;
    mesh2_skin.morphTargetInfluences[mt2-1] = 1;
} else {
    mesh1_bone.morphTargetInfluences[mt1-1] = 0;
    mesh1_bone.morphTargetInfluences[mt2-1] = 1;
    mesh2_skin.morphTargetInfluences[mt1-1] = 0;
    mesh2_skin.morphTargetInfluences[mt2-1] = 1;
}
StopAnimation -= 1;

if(canMorph) {
    controls2.target.y = targetArray[mt2];
    camera2.position.x = targetArray3D[mt2][0];
    camera2.position.y = targetArray3D[mt2][1];
    camera2.position.z = targetArray3D[mt2][2];
}

} else if (Animation == false && 0 <= mesh1_bone.morphTargetInfluences[mt2-1] && SliderMoveR ==false) {
    if(mt1==0){
        mesh1_bone.morphTargetInfluences[mt2-1] -= AnimationSpeed;
        mesh2_skin.morphTargetInfluences[mt2-1] -= AnimationSpeed;

    } else {
        mesh1_bone.morphTargetInfluences[mt2-1] -= AnimationSpeed;
        mesh1_bone.morphTargetInfluences[mt1-1] += AnimationSpeed;;
        mesh2_skin.morphTargetInfluences[mt2-1] -= AnimationSpeed;
        mesh2_skin.morphTargetInfluences[mt1-1] += AnimationSpeed;;
    }

}

ErrCounter = 0;

if(canMorph){
    if(targetErr3D[0] == false) {
        camera2.position.x -= targetDelta3D[0]/40;
        camera2.position.z -= targetDelta3D[2]/40;
    } else {
        camera2.position.x += targetDelta3D[0]/40;
        camera2.position.z += targetDelta3D[2]/40;
    }
    if(targetErr3D[1] == false) {
        controls2.target.y -= targetDelta/40;
        camera2.position.y -= targetDelta3D[1]/40;
    } else {
        controls2.target.y += targetDelta/40;
        camera2.position.y += targetDelta3D[1]/40;
    }
}

} else if (mesh1_bone.morphTargetInfluences[mt2-1] < 0){
    Animation = true;
    SliderMoveR = true;
    mesh1_bone.morphTargetInfluences[mt2-1] = 0;
    mesh2_skin.morphTargetInfluences[mt2-1] = 0;
    StopAnimation -= 1;

    if(canMorph) {
        controls2.target.y = targetArray[mt1];
        camera2.position.x = targetArray3D[mt1][0];
        camera2.position.y = targetArray3D[mt1][1];
        camera2.position.z = targetArray3D[mt1][2];
    }
}

} else if (StopAnimation != StopTime){
    if (StopAnimation == 0) {
        StopAnimation = StopTime;
    } else if (0 < StopAnimation) {
        StopAnimation -= 1;
    }
}
}

```



```

rV = mesh1_bone.morphTargetInfluences[mt2-1];

if (AutoRotation == true) {
    mesh1_bone.rotation.y += -0.5 *(Math.PI / 360);
    mesh2_skin.rotation.y += -0.5 *(Math.PI / 360);
    lineX.rotation.y += -0.5 *(Math.PI / 360);
    lineY.rotation.y += -0.5 *(Math.PI / 360);
    lineZ.rotation.y += -0.5 *(Math.PI / 360);
    Texts3.rotation.y += -0.5 *(Math.PI / 360);

} else {
    if(canMorph == true){
        if(counter != 0)
            moveCounter++;
        if(moveCounter > 500) {
        }
    }
}

render();
}

function setRange(value) {
    document.getElementById("morphRange").value = value;

    if(document.getElementById("morphRange").value == 0.5){
        ErrCounter += 1;
        if(ErrCounter > 1) {
            var valueStr = Number(value.substring(0,4));
            if(mt1 == 0) {
                mesh1_bone.morphTargetInfluences[mt2-1] = valueStr;
                mesh2_skin.morphTargetInfluences[mt2-1] = valueStr;
            } else {
                mesh1_bone.morphTargetInfluences[mt1-1] = 1-valueStr;
                mesh2_skin.morphTargetInfluences[mt1-1] = 1-valueStr;
                mesh1_bone.morphTargetInfluences[mt2-1] = valueStr;
                mesh2_skin.morphTargetInfluences[mt2-1] = valueStr;
            }
            rV = valueStr;
            document.getElementById("morphRange").value = valueStr;
        }
    }
}

function canMorph3D(morphTarget1, morphTarget2) {
    mt1 = morphTarget1;
    mt2 = morphTarget2;
    for(var i=0; i < 7; i++) {
        if(mt1 == 0) {
            mesh1_bone.morphTargetInfluences[i] = 0;
            mesh2_skin.morphTargetInfluences[i] = 0;
        } else {
            if(i == mt1) {
                mesh1_bone.morphTargetInfluences[i-1] = 1;
                mesh2_skin.morphTargetInfluences[i-1] = 1;
            } else {
                mesh1_bone.morphTargetInfluences[i-1] = 0;
                mesh2_skin.morphTargetInfluences[i-1] = 0;
            }
        }
    }
}

rV = 0;

counter++;

if(counter == 1) {
    $("#btn3-1").css({opacity: "1.0"});
    $("#btn3-2").css({opacity: "1.0"});
    setAutoMorphing();
}

```

```

        setAutoRotation();
        Colorize = false;
        // changeTexture();
        controls.noPan = false;
        controls.noZoom = false;
        controls.noRotate = false;
    }

    if(Colorize == true) {
        $("#btn3-1").css({opacity: "1.0"});
        mesh1_bone.material.map = new THREE.ImageUtils.loadTexture("model/Skeleton_path6/0" + mt1 + "-0" + mt2 + ".png");
    }

    controls2.target.y = targetArray[mt1];
    targetDef = targetArray[mt1];
    targetDelta = Math.floor( Math.max(targetArray[mt1], targetArray[mt2]) - Math.min(targetArray[mt1], targetArray[mt2]));

    for(var i=0; i<3; i++) {
        targetDef3D[i] = targetArray3D[mt1][i];
        targetDelta3D[i] = Math.floor( Math.max(targetArray3D[mt1][i], targetArray3D[mt2][i]) - Math.min(targetArray3D[mt1][i],
targetArray3D[mt2][i]));
    }
    camera2.position.x = targetDef3D[0];
    camera2.position.y = targetDef3D[1];
    camera2.position.z = targetDef3D[2];

    if(targetArray3D[mt1][0]>targetArray3D[mt2][0]) {
        targetErr3D[0] = true;
    } else {
        targetErr3D[0] = false;
    }

    if(targetArray3D[mt1][1]>targetArray3D[mt2][1]) {
        targetErr3D[1] = true;
    } else {
        targetErr3D[1] = false;
    }

    if(targetArray3D[mt1][2]>targetArray3D[mt2][2]) {
        targetErr3D[2] = true;
    } else {
        targetErr3D[2] = false;
    }
}

function cannotMorph3D() {
    setRange(0.5);
}

var controls = new THREE.OrbitControls(camera);
controls.target = new THREE.Vector3(0, 0, 0);
controls.noPan = true;
controls.noZoom = true;
controls.noRotate = true;

var controls2 = new THREE.OrbitControls(camera2);
controls2.target = new THREE.Vector3(0, 0, 0);
controls2.noPan = true;
controls2.noZoom = true;
controls2.noRotate = true;

var controlsD = new THREE.OrbitControls(cameraD);
controlsD.target = new THREE.Vector3(0, 0, 0);

// Overlaid rendering
function render() {

    if(panCounter == 0) {
        panCounter = 1;

```

```

    }
    controls.update();
    controls2.update();
    controlsD.update();
    renderer.clear();

    if(SkeletonComparing) {
        renderer.render(scene3,camera2);
        if(canMorph == true) {
            setRange(rV);
            renderer.clearDepth();
            renderer.render(scene2,camera2);
            renderer.clearDepth();
            renderer.render(scene,camera2);
        }
    } else {
        renderer.render(scene3,camera);
        if(canMorph == true) {
            setRange(rV);
            renderer.clearDepth();
            renderer.render(scene2,camera);
            renderer.clearDepth();
            renderer.render(scene,camera);
        }
    }
}

// Detarminate touch / mouse position
var mousedown = false;

document.body.addEventListener('mousedown', function(e){
    if((e.pageX > 100 && e.pageX < 260) && (e.pageY > 120 && e.pageY < 140)) {
        controls.enabled = false;
        controls2.enabled = false;
        controlsD.enabled = false;
    } else {
        controls.enabled = true;
        controls2.enabled = true;
        controlsD.enabled = true;
    }
    mousedown = true;
    prevPosition = {x: e.pageX, y: e.pageY};
}, false);

renderer.domElement.addEventListener('mousedown', function(e){
    controls.enabled = true;
    controls2.enabled = true;
    controlsD.enabled = true;
}, false);

renderer.domElement.addEventListener('mousemove', function(e){
    if(!mousedown) return;

    if(SkeletonComparing && controls.enabled) {
        moveDistance = {x: prevPosition.x - e.pageX, y: prevPosition.y - e.pageY};

        mesh1_bone.rotation.y -= moveDistance.x * 0.01;
        mesh2_skin.rotation.y -= moveDistance.x * 0.01;
        lineX.rotation.y -= moveDistance.x * 0.01;
        lineY.rotation.y -= moveDistance.x * 0.01;
        lineZ.rotation.y -= moveDistance.x * 0.01;
        Texts3.rotation.y -= moveDistance.x * 0.01;
    }

    prevPosition = {x: e.pageX, y: e.pageY};
}, false);

```

```

renderer.domElement.addEventListener('mouseup', function(e){
  mousedown = false;
  controls.enabled = true;
  controls2.enabled = true;
  controlsD.enabled = true;
}, false);

document.body.addEventListener('touchstart', function(e){
  if((e.pageX > 100 && e.pageX < 260) && (e.pageY > 120 && e.pageY < 140)) {
    controls.enabled = false;
    controls2.enabled = false;
    controlsD.enabled = false;
  } else {
    controls.enabled = true;
    controls2.enabled = true;
    controlsD.enabled = true;
  }
  mousedown = true;
  prevPosition = {x: e.pageX, y: e.pageY};
}, false);

renderer.domElement.addEventListener('touchstart', function(e){
  controls.enabled = true;
  controls2.enabled = true;
  controlsD.enabled = true;
}, false);

renderer.domElement.addEventListener('touchmove', function(e){
  if(!mousedown) return;

  if(SkeletonComparing && controls.enabled) {
    moveDistance = {x: prevPosition.x - e.pageX, y: prevPosition.y - e.pageY};

    mesh1_bone.rotation.y -= moveDistance.x * 0.01;
    mesh2_skin.rotation.y -= moveDistance.x * 0.01;
    lineX.rotation.y -= moveDistance.x * 0.01;
    lineY.rotation.y -= moveDistance.x * 0.01;
    lineZ.rotation.y -= moveDistance.x * 0.01;
    Texts3.rotation.y -= moveDistance.x * 0.01;
  }

  prevPosition = {x: e.pageX, y: e.pageY};
}, false);

renderer.domElement.addEventListener('touchend', function(e){
  mousedown = false;
  controls.enabled = true;
  controls2.enabled = true;
  controlsD.enabled = true;
}, false);

</script>

</body>
</html>

```

```

html {
  overflow: auto;
  width: 100%;
}

body {
  font-family: sans-serif;
  background-color: #000000;
  margin: 0px;
  overflow: hidden;
  color: rgba(200, 200, 200, 1.0);
}

a: hover img {
  opacity: 0.7;
  filter: alpha(opacity=70);
}

h1 {
  font-size: 25px;
  padding: 0;
  margin: 0;
  text-align: center;
}

h2 {
  font-size: 20px;
  padding: 0;
  margin: 0;
  text-align: center;
  margin-bottom: -5px;
}

#ctrl {
  display: block;
  float: left;
  position: absolute;
  top: 20px;
  left: 20px;
  height: 310px;
  width: 280px;
  text-align: left;
  padding: 10px 20px;
  font-size: 13px;
  background-color: rgba(230, 230, 230, 0.15);
  border: thin solid #666666;
  border-radius: 1px;
  z-index: 100;
}

#sliderControl {
  margin-top: 20px;
  margin-bottom: 30px;
  height: 550px;
  display: block;
  float: left;
}

#rangelmage1 {
  float: left;
  display: block;
  height: 50px;
  width: 50px;
  background-color: rgba(230, 230, 230, 0.10);
}

#morphRange {
  float: left;
  display: block;
  width: 159px;
  position: relative;
  top: 15px;
  margin-left: 8px;
  margin-right: 8px;
}

#rangelmage2 {
  float: left;
  display: block;
  height: 50px;
  width: 50px;
  background-color: rgba(230, 230, 230, 0.10);
}

#btn1 {
  float: left;
  display: block;
  margin-top: 20px;
  margin-bottom: 10px;
  margin-right: 10px;
  margin-bottom: 10px;
  opacity: 0.4;
}

#btn2 {
  float: left;
  display: block;
  margin-top: 20px;
  margin-bottom: 10px;
  opacity: 0.4;
}

.btn3 {
  float: left;
  display: block;
  margin-right: 13px;
  margin-bottom: 10px;
}

.btn3-2 {
  float: left;
  display: block;
}

#btn3-1, #btn3-2, #btn3-3 {
  opacity: 0.4;
}

#gt {
  float: left;
  display: block;
  width: 280px;
  position: absolute;
  bottom: 20px;
  left: 20px;
  padding: 0px 20px;
  margin-top: 30px;
  border: thin solid #666666;
  border-radius: 1px;
  background-color: rgba(230, 230, 230, 0.15);
}

```

```

        z-index: 1000;
    }
    #GT_Main {
        float: left;
        display: block;
        width: 285px;
        height: 290px;
    }

    #GT_1 {
        float: left;
        display: block;
        width: 32px;
        height: 101px;
        padding-top: 85px;
    }

    #GT_2 {
        float: left;
        display: block;
        width: 180px;
        height: 170px;
    }

    #GT_3 {
        float: left;
        display: block;
        width: 32px;
        height: 203px;
        margin-top: 10px;
    }

    #GT_4 {
        float: left;
        display: block;
        width: 32px;
        height: 170px;
    }

    #GT_5 {
        float: left;
        display: block;
        width: 285px;
        height: 33px;
    }

    #Arw {
        padding-top: 0px;
    }

    #Icon1, #Icon2, #Icon3, #Icon4, #Icon5, #Icon6 {
        margin-left: 4px;
        margin-top: -4px;
    }

    #Icon0:hover, #Icon1:hover, #Icon2:hover, #Icon3:hover,
    #Icon4:hover, #Icon5:hover, #Icon6:hover {
        opacity: 0.5;
    }

```

```

var Selector1 = -1;
var Selector2 = -1;
var canMorph = false;

$(document).ready(function(){
    console.info("Selector1:", Selector1);
    console.info("Selector2:", Selector2);
})

function selectPath(Selected) {

    var SelectedTmp1 = Selected;

    if (Selector1 + Selector2 == -2) { //ひとつも選択されていない場合
        Selector1 = SelectedTmp1;
        $('#Icon'+Selected).attr("src","img/gt2/Icon" + Selected + ".v.png");
        $('#Mes').attr("src","img/gt2/Mes_p1.png");
        $('#Arw').css('padding-top', '0px');
        $('#Arw').attr("src","img/gt2/A0.png");
        canMorph = false;
    } else if (Selector1 == -1 || Selector2 == -1) { //片方だけ選択されている場合
        if (Selector2 == -1) {
            if(Selector1 == Selected) { //一つ選択済み->取り消し Selected1
                Selector1 = -1;
                $('#Icon'+Selected).attr("src","img/gt2/Icon" + Selected + ".g.png");
                $('#rangelImage1').attr("src","img/gt2/lconx.png");
                $('#rangelImage2').attr("src","img/gt2/lconx.png");
                $('#Mes').attr("src","img/gt2/Mes_p2.png");
                $('#Arw').css('padding-top', '0px');
                $('#Arw').attr("src","img/gt2/A0.png");
                $('#Path').attr("src","img/gt2/Path6_0.png");
                canMorph = false;
            } else { //一つ選択済み->追加 Selected1
                Selector2 = Selected;
                if(Selector1 == 0 || Selector2 == 0) { //どちらかが 0->進化
                    $('#Mes').attr("src","img/gt2/Mes_e.png");
                    $('#Icon'+Selector1).attr("src","img/gt2/Icon" + Selector1 + ".p.png");
                    $('#Icon'+Selector2).attr("src","img/gt2/Icon" + Selector2 + ".p.png");
                    $('#rangelImage1').attr("src","img/gt2/Icon" + Math.min(Selector1,Selector2) + ".p.png");
                    $('#rangelImage2').attr("src","img/gt2/Icon" + Math.max(Selector1,Selector2) + ".p.png");
                    $('#Path').attr("src","img/gt2/Path6_0-" + Math.max(Selector1, Selector2) + ".png");
                    $('#Arw').css('padding-top', '0px');
                    $('#Arw').attr("src","img/gt2/A0.png");
                    canMorph = true;
                } else { //どちらも 0 でない->比較
                    $('#Mes').attr("src","img/gt2/Mes_c.png");
                    $('#Icon'+Selector1).attr("src","img/gt2/Icon" + Selector1 + ".b.png");
                    $('#Icon'+Selector2).attr("src","img/gt2/Icon" + Selector2 + ".b.png");
                    $('#rangelImage1').attr("src","img/gt2/Icon" + Math.min(Selector1,Selector2) + ".b.png");
                    $('#rangelImage2').attr("src","img/gt2/Icon" + Math.max(Selector1,Selector2) + ".b.png");
                    $('#Path').attr("src","img/gt2/Path6_0.png");
                    var pdT = (Math.min(Selector1, Selector2)-1)*34+5;
                    $('#Arw').css('padding-top', pdT + 'px');
                    $('#Arw').attr("src","img/gt2/A" + Math.abs(Selector1-Selector2) + ".png");
                    canMorph = true;
                }
            }
        } else if (Selector1 == -1) {
            if(Selector2 == Selected) { //一つ選択済み->取り消し Selected1
                Selector1 = -1;
                $('#Icon'+Selected).attr("src","img/gt2/Icon" + Selected + ".g.png");
                $('#rangelImage1').attr("src","img/gt2/lconx.png");
                $('#rangelImage2').attr("src","img/gt2/lconx.png");
                $('#Mes').attr("src","img/gt2/Mes_p2.png");
                $('#Arw').css('padding-top', '0px');
            }
        }
    }
}

```



```

    $('#Arw').attr("src","img/gt2/A0.png");
    $('#Path').attr("src","img/gt2/Path6_0.png");
    canMorph = false;
  } else { //一つ選択済み->追加 Selected1
    Selector1 = Selected;
    if(Selector1 == 0 || Selector2 == 0) { //どちらかが 0->進化
      $('#Mes').attr("src","img/gt2/Mes_e.png");
      $('#Icon'+Selector1).attr("src","img/gt2/Icon" + Selector1 + ".png");
      $('#Icon'+Selector2).attr("src","img/gt2/Icon" + Selector2 + ".png");
      $('#rangelImage1').attr("src","img/gt2/Icon" + Math.min(Selector1,Selector2) + ".png");
      $('#rangelImage2').attr("src","img/gt2/Icon" + Math.max(Selector1,Selector2) + ".png");
      $('#Path').attr("src","img/gt2/Path6_0-" + Math.max(Selector1, Selector2) + ".png");
      $('#Arw').css('padding-top', '0px');
      $('#Arw').attr("src","img/gt2/A0.png");
      canMorph = true;
    } else { //どちらも 0 でない->比較
      $('#Mes').attr("src","img/gt2/Mes_c.png");
      $('#Icon'+Selector1).attr("src","img/gt2/Icon" + Selector1 + ".b.png");
      $('#Icon'+Selector2).attr("src","img/gt2/Icon" + Selector2 + ".b.png");
      $('#rangelImage1').attr("src","img/gt2/Icon" + Math.min(Selector1,Selector2) + ".b.png");
      $('#rangelImage2').attr("src","img/gt2/Icon" + Math.max(Selector1,Selector2) + ".b.png");
      var pdT = (Math.min(Selector1, Selector2)-1)*34+5;
      $('#Arw').css('padding-top', pdT + 'px');
      $('#Arw').attr("src","img/gt2/A" + Math.abs(Selector1-Selector2) + ".png");
      canMorph = true;
    }
  }
}
} else { //両方選択されている場合
  if (Selector1 == Selected) { //片方取り消し #1
    Selector1 = -1;
    $('#Icon'+Selected).attr("src","img/gt2/Icon" + Selected + ".g.png");
    $('#Icon'+Selector2).attr("src","img/gt2/Icon" + Selector2 + ".v.png");
    $('#rangelImage1').attr("src","img/gt2/Iconx.png");
    $('#rangelImage2').attr("src","img/gt2/Iconx.png");
    $('#Mes').attr("src","img/gt2/Mes_p1.png");
    $('#Arw').css('padding-top', '0px');
    $('#Arw').attr("src","img/gt2/A0.png");
    $('#Path').attr("src","img/gt2/Path6_0.png");
    canMorph = false;
  } else if (Selector2 == Selected) { //片方取り消し #2
    Selector2 = -1;
    $('#Icon'+Selected).attr("src","img/gt2/Icon" + Selected + ".g.png");
    $('#Icon'+Selector1).attr("src","img/gt2/Icon" + Selector1 + ".v.png");
    $('#rangelImage1').attr("src","img/gt2/Iconx.png");
    $('#rangelImage2').attr("src","img/gt2/Iconx.png");
    $('#Mes').attr("src","img/gt2/Mes_p1.png");
    $('#Arw').css('padding-top', '0px');
    $('#Arw').attr("src","img/gt2/A0.png");
    $('#Path').attr("src","img/gt2/Path6_0.png");
    canMorph = false;
  } else if (Selector1 == Selector2) { //両方同じ値になってしまった場合
    Selector2 = -1;
    $('#Icon'+Selected).attr("src","img/gt2/Icon" + Selected + ".g.png");
    $('#Icon'+Selector1).attr("src","img/gt2/Icon" + Selector1 + ".v.png");
    $('#rangelImage1').attr("src","img/gt2/Iconx.png");
    $('#rangelImage2').attr("src","img/gt2/Iconx.png");
    $('#Mes').attr("src","img/gt2/Mes_p1.png");
    $('#Arw').css('padding-top', '0px');
    $('#Arw').attr("src","img/gt2/A0.png");
    $('#Path').attr("src","img/gt2/Path6_0.png");
    canMorph = false;
  } else if ((Selector1 != -1 && Selector2 != -1) && (Selector1 != Selected && Selector2 != Selected)) { //3 つめが選択
    された場合
    $('#Icon'+Selector1).attr("src","img/gt2/Icon" + Selector1 + ".g.png");

```

```

    $('#Icon'+Selector2).attr("src","img/gt2/Icon" + Selector2 + ".png");
    $('#rangelImage1').attr("src","img/gt2/lconx.png");
    $('#rangelImage2').attr("src","img/gt2/lconx.png");
    Selector1 = Selected;
    Selector2 = -1;
    $('#Icon'+Selector1).attr("src","img/gt2/Icon" + Selector1 + ".v.png");
    $('#Mes').attr("src","img/gt2/Mes_p1.png");
    $('#Arw').css('padding-top', '0px');
    $('#Arw').attr("src","img/gt2/A0.png");
    $('#Path').attr("src","img/gt2/Path6_0.png");
    canMorph = false;
  }
}

console.log("canMorph: ", canMorph);
console.log("Selector: ", Selector1, Selector2);

if (canMorph == true) {
  document.getElementById("MorphTarget1").value = Math.min(Selector1, Selector2);
  document.getElementById("MorphTarget2").value = Math.max(Selector1, Selector2);
  console.log("MorphTarget: ", document.getElementById("MorphTarget1").value, " , ",
document.getElementById("MorphTarget2").value);
  canMorph3D(document.getElementById("MorphTarget1").value,
document.getElementById("MorphTarget2").value);
  // モデルのロードへ
} else {
  cannotMorph3D();
}
document.onkeydown = pageMove;
}

function Clear() {
  var Selector1 = -1;
  var Selector2 = -1;
  console.log("clear().");
  console.info("Selector1:", Selector1);
  console.info("Selector2:", Selector2);
}

```

```

{
  "metadata" :
  {
    "formatVersion" : 3.1,
    "sourceFile" : "Skeleton.obj",
    "generatedBy" : "OBJConverter",
    "vertices" : 3314,
    "faces" : 3474,
    "normals" : 0,
    "colors" : 0,
    "uvs" : 3314,
    "materials" : 0
  },

  "scale" : 1.000000,

  "materials": [ {
    "DbgColor" : 15658734,
    "DbgIndex" : 0,
    "DbgName" : "default"
  } ],

  "vertices": [0.046838,0.427194,0.362788, (省略 : ベースモデルの頂点データ)],

  "morphTargets": [
    { "name": "Skeleton_01", "vertices": [0.374954,7.092189, (省略 : モーフターゲット 1 の頂点データ)] },
    { "name": "Skeleton_02", "vertices": [0.198713,2.292710, (省略 : モーフターゲット 2 の頂点データ)] },
    { "name": "Skeleton_03", "vertices": [0.161443,2.928966, (省略 : モーフターゲット 3 の頂点データ)] },
    { "name": "Skeleton_04", "vertices": [0.160328,4.126679, (省略 : モーフターゲット 4 の頂点データ)] },
    { "name": "Skeleton_05", "vertices": [0.400236,7.795532, (省略 : モーフターゲット 5 の頂点データ)] },
    { "name": "Skeleton_06", "vertices": [1.108540,13.354311, (省略 : モーフターゲット 6 の頂点データ)] },
  ]

  "morphColors": [],

  "normals": [],

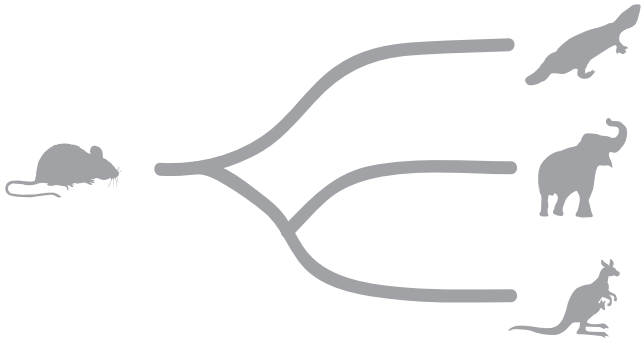
  "colors": [],

  "uvs": [[0.26066,0.84232, (省略 : UV マップデータ)],

  "faces": [11,56,( 省略 : 面の構成データ)]
}

```

系統樹について



系統樹とは、生物の進化の流れを一本の樹木に見立てて描いた図のことです。

左図の場合、哺乳類が共通の祖先から進化の過程で分岐し、種を増やしていったことが示されています。この図から、ゾウとカンガルーは、カモノハシよりも近い関係の種であるということがわかります。

今回扱う生物の種について

今回教材で閲覧していただく生物は、以下の哺乳類です。



アデロバシレウス

(発見されている最古の哺乳類)



ヒト



ウマ



イヌ



ネコ



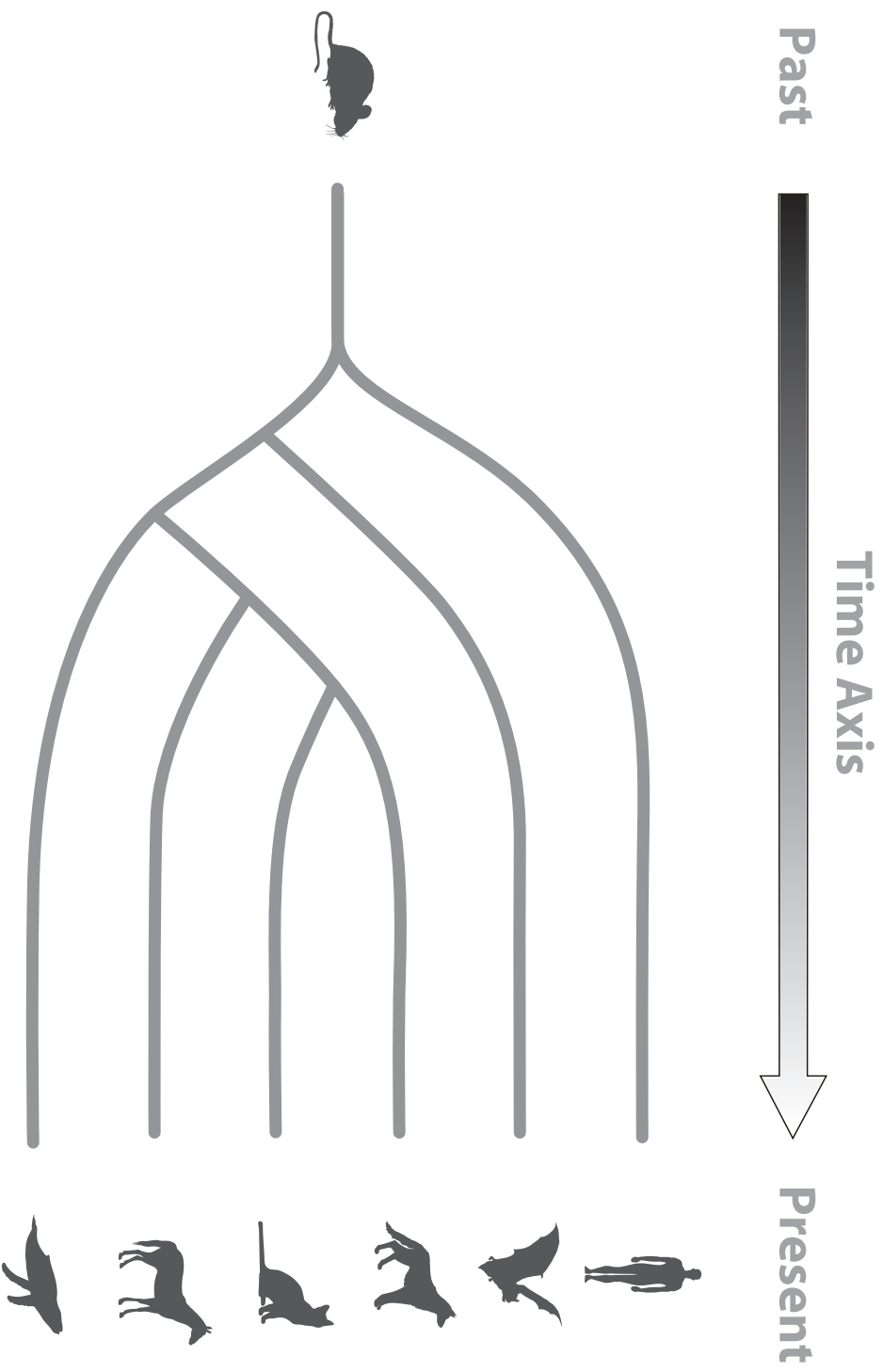
コウモリ



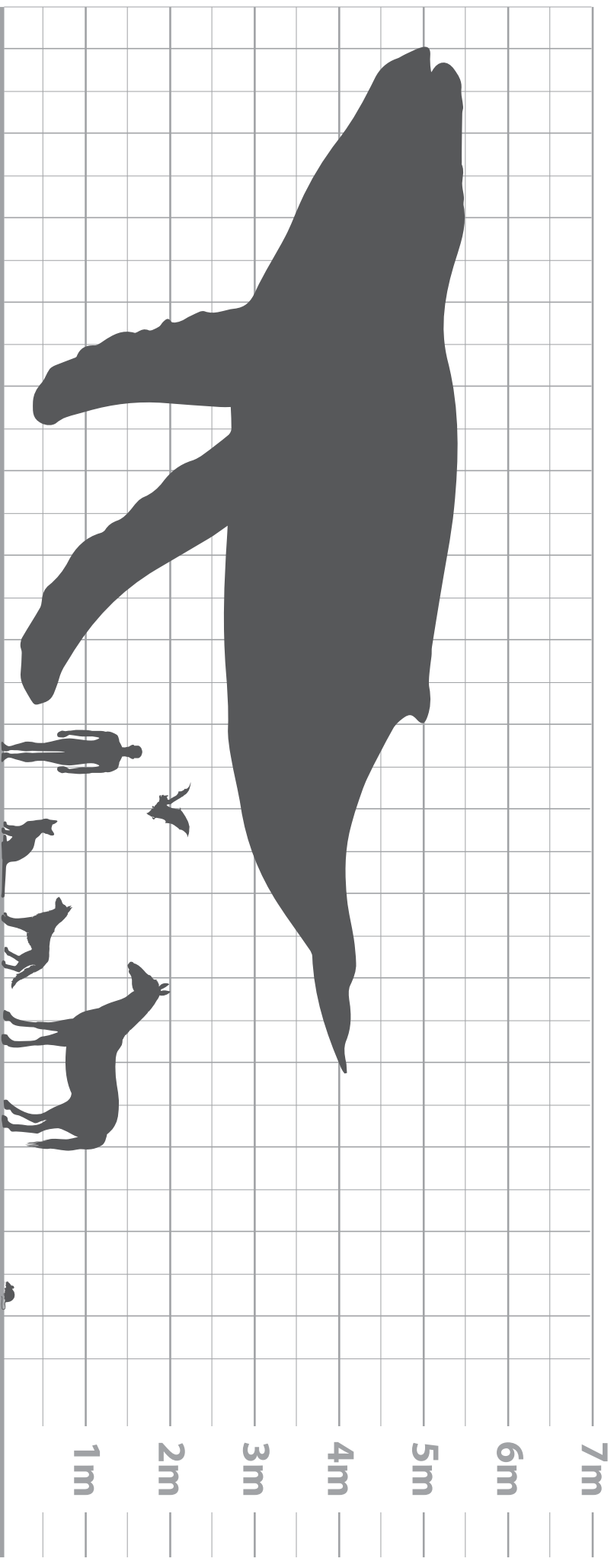
クジラ

現在生きている生物

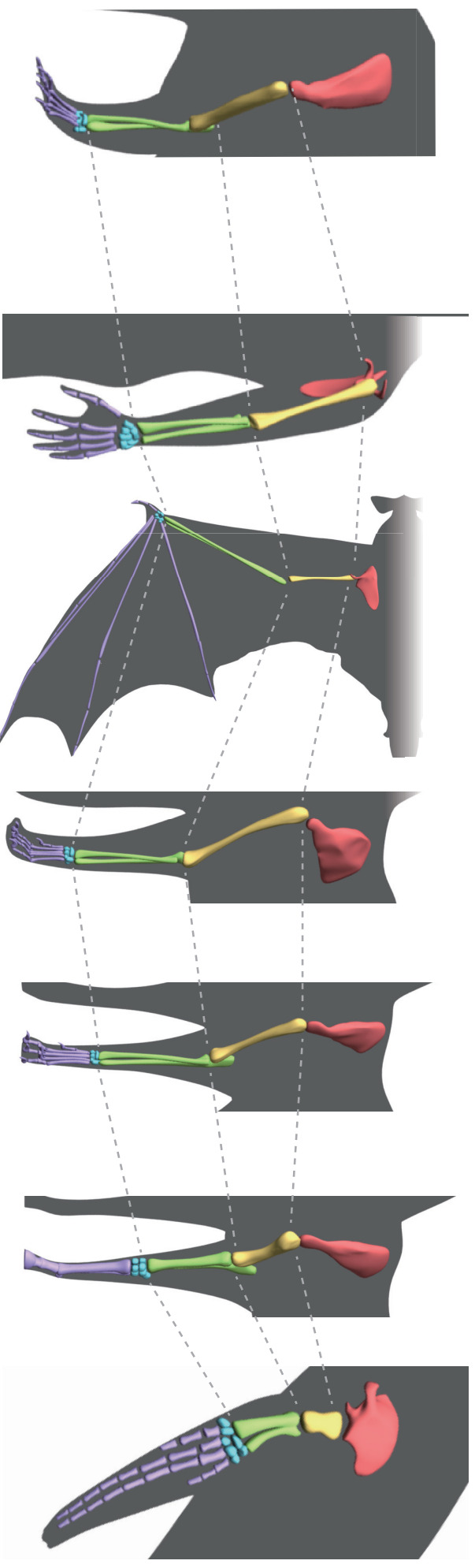
👁 Phylogenetic tree



Size



Skeleton



教材に関するテスト

事前に実験教材で閲覧していただいた、

アデロバシレウス (哺乳類の祖先)、ヒト、コウモリ、ネコ、イヌ、ウマ、クジラについて、以下の設問にお答えください。

教材の内容から判断できないと感じた設問は、飛ばしても構いません。

※次のページへ進んだら、前のページには戻らないようにしてください。

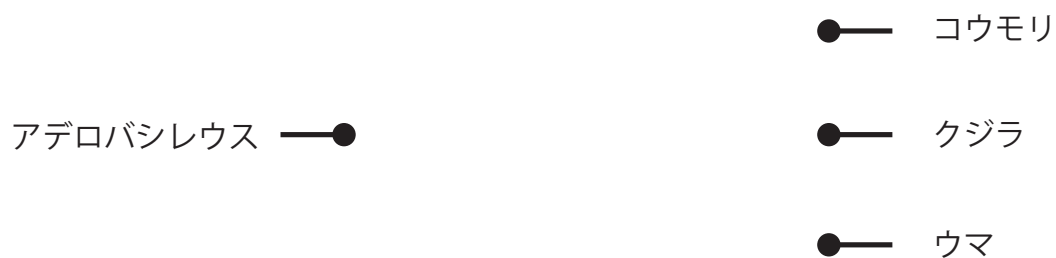
- ①ネコ、ヒト、イヌ、ウマを、一般的な体長 (ヒトの場合は身長) の大きい順に並べ替えてください。

> > >

- ②以下の生物から、5本指の種をすべて選択し、該当する選択肢に○をつけてください。

クジラ ヒト イヌ ネコ アデロバシレウス コウモリ ウマ

- ③下図の ● を線でつなぎ、アデロバシレウス、コウモリ、ウマ、クジラについての系統樹を完成させてください。



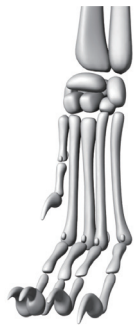
- ④以下の生物の中で、最も指骨 (指を形成している骨) の数が多い種を選択し、該当する選択肢に○をつけてください。

アデロバシレウス コウモリ ネコ ヒト クジラ ウマ イヌ

⑤以下の生物から、イヌと進化学的に近いと考えられる種を3つ選択し、該当する選択肢に○をつけてください。

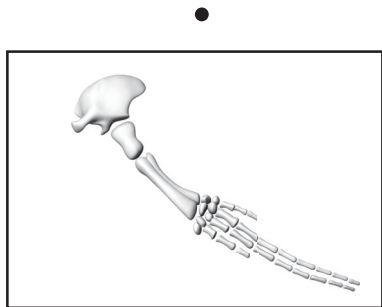
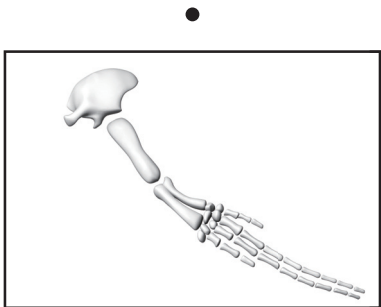
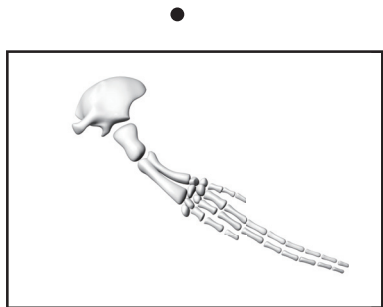
ヒト	クジラ	ウマ	コウモリ	ネコ
----	-----	----	------	----

⑥下図はどの生物の骨格でしょうか？該当する選択肢に○をつけてください。

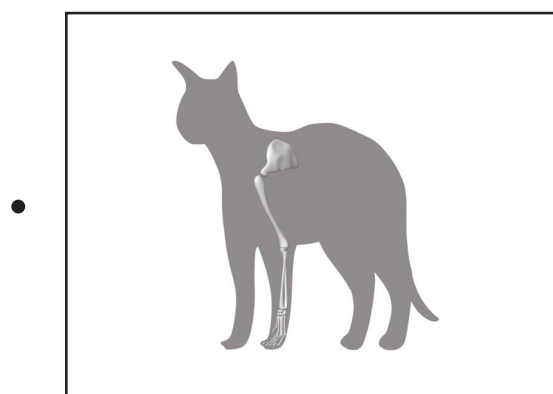
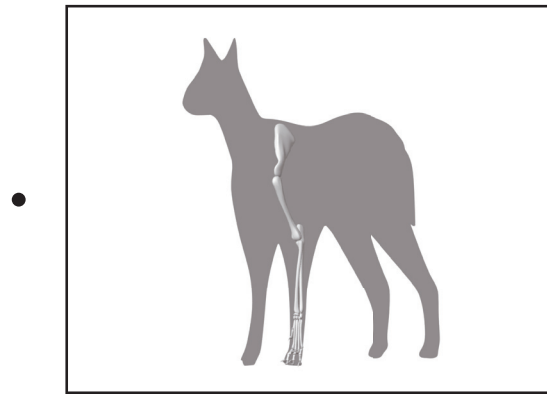
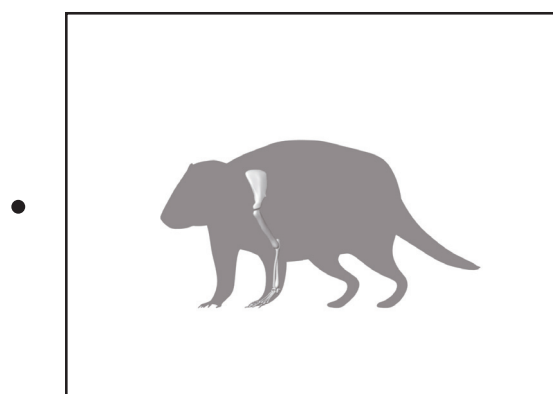
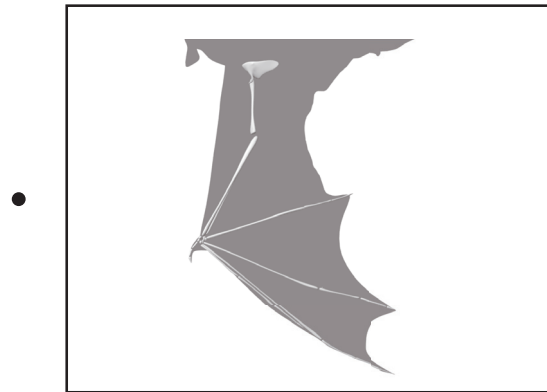
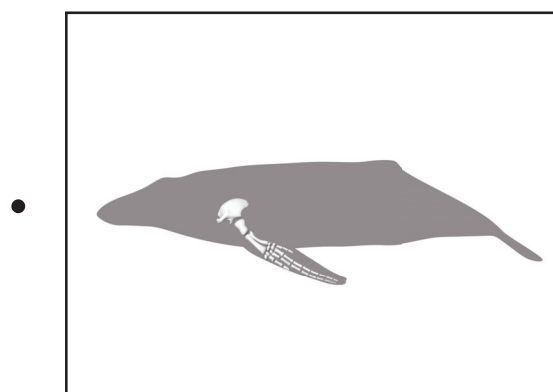
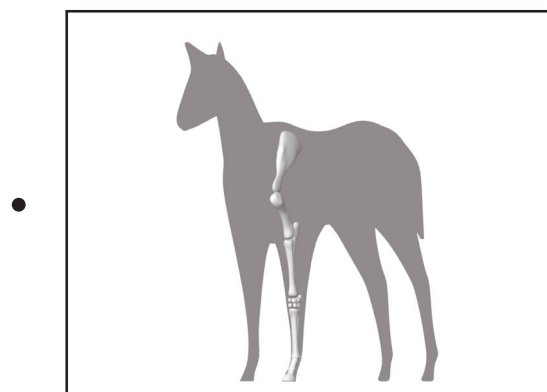
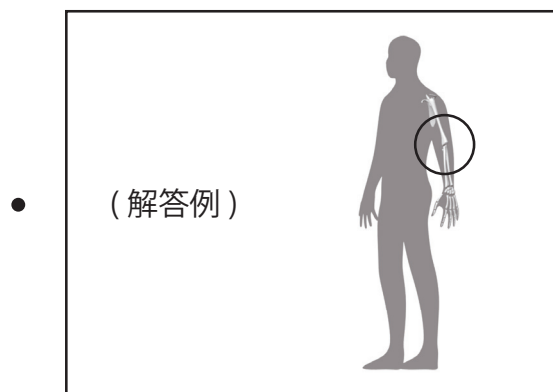


ネコ	イヌ	ウマ
アデロバシレウス	ヒト	コウモリ

⑦下図から、クジラの前肢骨格を正しく表した図を選択し、該当する選択肢に○をつけてください。



⑧以下の画像それぞれについて、ひじであると考えられる場所に○をつけてください。



(解答例)

⑨以下の図は全て、生物の左前肢の全体もしくは一部分を表しています。

左図で赤く示された部分は、それぞれ右図のどの部分に該当するでしょうか？該当する部分に○をつけてください。

